

# **CURRICULUM REVISION PROJECT**

**2012**

## **TEACHER GUIDE FOR**

Advanced Java Programming  
17625

## **SIXTH SEMESTER COMPUTER ENGINEERING GROUP**

**December 2014**



**MAHARASHTRA STATE  
BOARD OF TECHNICAL EDUCATION, Mumbai**  
(Autonomous) (ISO 9001:2008) (ISO/IEC 27001:2005)

## INDEX

<b>Content No.</b>	<b>Contents</b>	<b>Page No.</b>
1.0	APPROACH TO CURRICULUM DESIGN	3
2.0	OBJECTIVES	8
3.0	CONTENT ANALYSIS	13
4.0	CURRICULUM	20
5.0	IMPLIMENTATION STRATEGIES	26
5.1	Planning of Lectures for a Semester with Content Detailing	26
5.2	Planning and Conduct of Test	55
5.3	Detail about Conduct of Assignment	55
5.4	Strategies for conduct of Practical	56
6.0	MODE OF ASSESSMENT	57
6.1	Class Test	57
6.1.1	Sample Test Paper –I	58
6.1.2	Sample Test Paper –II	70
6.2	Sample Question Paper	78
6.2.1	Specification Table	106
6.2.2	Question Paper Profile	107
6.2.3	Answer Key	108

## 1.0 APPROACH TO CURRICULUM DESIGN

### 1.1 Background:

MSBTE is introducing the revised curriculum under 'G' scheme from the academic year 2012-13.

There are many institutions in the state running different diploma courses. In order to ensure uniform and effective implementation of the curriculum it is necessary that every teacher is aware of approach for curriculum design, educational principles to be adopted, learning resources to be used and evaluation methods. The teacher guide prepared for each subject will provide the inputs related to above mentioned aspects to achieve uniform and effective implementation of curriculum of various subjects.

### 1.2 CURRICULUM PHILOSOPHY

MSBTE has adopted systems approach while designing the scientific based curriculum since 1995. The same approach has been adopted while revising the curriculum in semester pattern.

Fig. No. 1 shows the systems diagram. This diagram provides the holistic view for curriculum designing, development, implementation and evaluation

The input to polytechnic education system is the students having 10+ qualifications. The teaching learning process occurs in the institution for six/eight semesters. The output of the system i. e. Diploma pass out is normally the input to industries. (Some students do go for higher education). While designing the curriculum the expectations of the industries play a major role. Due to globalization and competition the industries expect that pass outs have generic and technological skills along with right attitude.

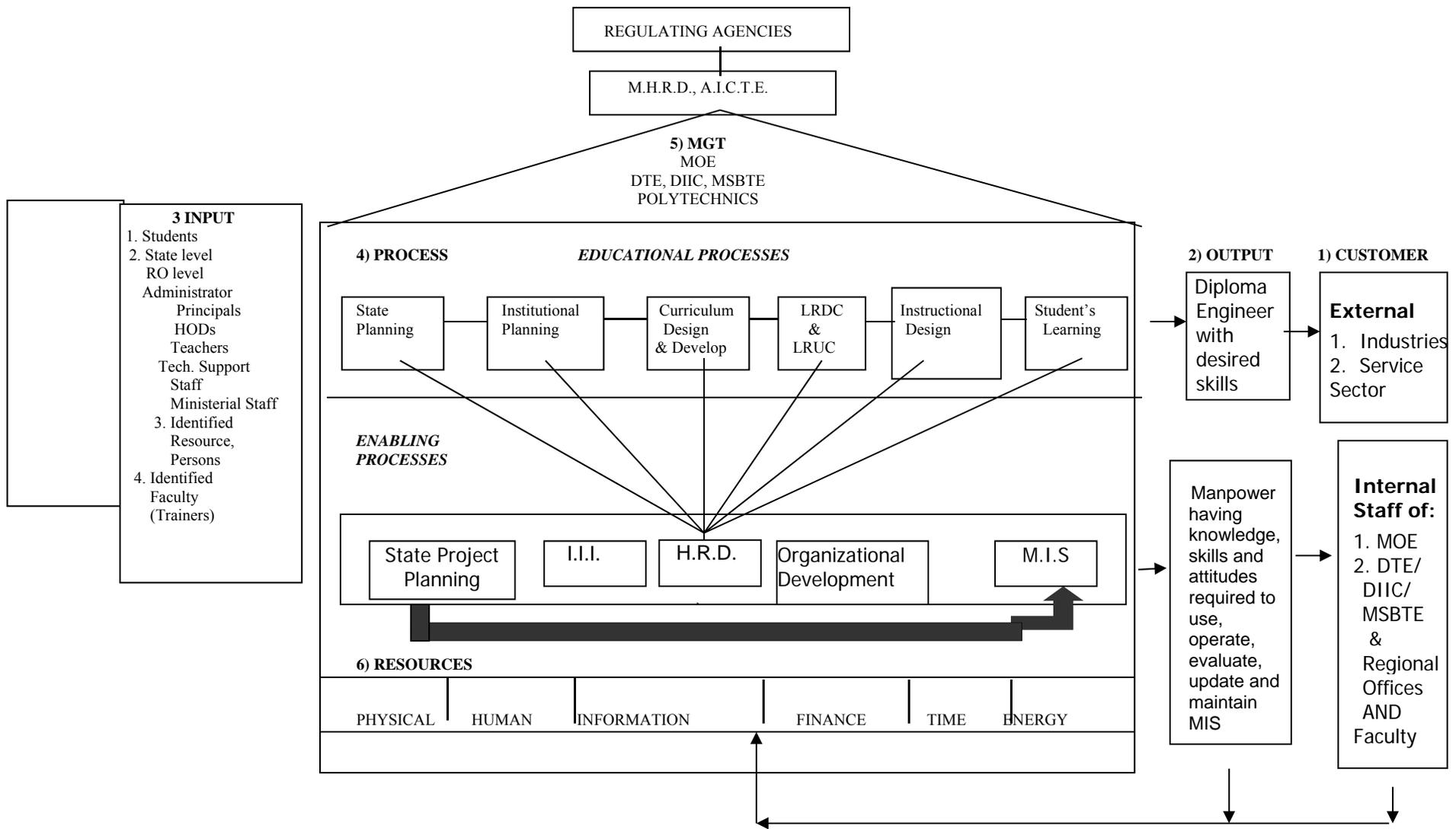
To fulfill the needs derived from systems approach following conceptual framework is considered:

### 1.3 Curriculum:

**“Curriculum is an educational program designed and implemented to achieve specified educational objectives”**

This definition takes into account the fact that

- Education is purposeful
- There is an organized plan of action contemplated
- Such a plan is translated into action through appropriate strategies of implementation.



**Fig 1 Systems Approach**

#### **1.4 Curriculum goals**

1. To develop confidence in students by providing more exposure to industry experience and world of work at global level
2. To provide conceptual knowledge and develop analytical ability
3. To develop communication skill with good English by providing sufficient practice
4. To enhance latest technical knowledge industry interaction and media
5. To develop learning to learn skills and life skills to cope up with industrial culture
6. To impart managerial skills by providing appropriate theoretical inputs
7. To develop problem solving ability through technical projects.

#### **1.5 DESIRED SKILLS**

Industries expect from the diploma engineer the abilities and skills of general nature and specific to the job performance. The curriculum aims at developing life skills and technological skills so that the diploma pass outs would be suitable for industry. The skills are listed below:

##### **Life Skills:**

- Search information from various sources
- Develop communication ability
- Develop Presentation skill
- Work as a member of a team/group and as leader
- Collect field data
- Develop Learning to learn
- Write report for given task/work/project
- Develop computer proficiency
- Develop observation skills

##### **Technological Skills:**

Diploma engineers should possess following intellectual and motor skills in order to satisfactorily perform duties assigned to them:

### **A) Intellectual skills.**

1. Identify the problem
2. Prepare the algorithms
3. Analyze the problem
4. Prepare the flowchart/model
5. Select hardware and software tools and technologies
6. Use of appropriate programming languages
7. Write programs
8. Test and debug computer Program
9. Diagnose the hardware faults
10. Prepare and interpret software documentation

### **B) Motor Skills.**

1. Handle the Computer system
2. Handling trouble shooting tools
3. Assemble and disassemble computer system
4. Install hardware devices
5. Install network

### **1.6 Salient Changes in the curriculum:**

- ❖ For First Semester Basic Science is divided into two parts- Basic Physics and Basic Chemistry. Theory examination of both parts as well as practical examination of both parts will be conducted on separate days. Sum of theory marks of both parts shall be considered for passing theory examination of Basic Science. Similarly it is applicable to practical examination. It is mandatory to appear for theory and practical examination of both parts. Candidate remaining absent in any examination of any section will not be declared successful for that exam head.
- ❖ For second semester Applied Science is divided into two sections- Applied Physics and Applied Chemistry where the theory examination of 50 marks each and practical examination of 25 Marks each will be conducted separately and the minimum passing marks for Applied Science will be the combination of both the sections. . It is mandatory

to appear for theory and practical examination of both parts. Candidate remaining absent in any examination of any section will not be declared successful for that exam head.

- ❖ The components of Development of Life Skills were taught in two semesters. In Development of Life Skills –I the topics related to personal development, such as Learning to Learn Skills, personality development, presentation skills etc. were included. In Development of Life Skills – II the topics related to Team Building, Leadership, group behavior etc. were covered. In the revised curriculum the scope of development of life skills has been broadened to include behavioral science component. Therefore the subject Development of Life Skills – II has been renamed and it is now included at Vth Semester in the revised curriculum under the title Behavioral Science.
- ❖ The subject of Professional Practices was introduced to integrate the skills acquired in Development of Life Skills, through technical subjects from second to sixth semester. The experience in implementing the contents of the subject shows that there are limited activities possible in second semester as the technical knowledge given to the students is very limited. Also at sixth semester the student are doing projects in which they are performing many activities included in the Professional Practices and therefore it is proposed that the subject of Professional Practices be prescribed only for three semesters viz. Third, fourth and fifth semesters.
- ❖ Introduction of Environment Studies at fourth Semester for all courses
- ❖ From the experience of implementation of Elective Subjects at V and VI semesters in last five years, it is proposed to have only one elective at the sixth semester for all courses. However the specialized courses like Medical Electronics, Electronics and Video Engineering will not have provision for electives. For elective, student will have to choose one from the given two/three subjects.
- ❖ While revising the curriculum redundant /obsolete topics/sub topics are being replaced by new/advance technology topics/sub topics.
- ❖ In Computer Engineering Group, for fourth Semester IF Computer Networks (CON) is replaced with Data Communication and Networking.
- ❖ For Fourth Semester IF, Applied Multimedia Technology Theory subject is changed to Practical.
- ❖ For Fifth semester CT, System Programming subject is included. For IF course, Information Security subject is included.

- ❖ For Sixth semester, elective subjects have been included.
- ❖ In order to satisfy the course objectives, online examination has been introduced for the subjects Management and Advanced Java Programming.
- ❖ Linux programming has been introduced as practical subject for CO/CM branch and Scripting Technology has been introduced as practical subject for IF.
- ❖ Mobile computing has been introduced for IF branch.

## **2.0 OBJECTIVES**

### **2.1 Introduction**

Objectives are the statements which describe the expected learning outcome. Such statements enable teachers to plan instructional process with appropriate resources. These objectives also provide a direction to frame proper questions to assess the learning outcome. During last decade there has been research on cognitive approach in psychology. This approach is based on biological structure of brain and meta-cognitive knowledge dimension. Important elements of this approach which form basics of learning are explained below.

### **2.2 Domains of Learning:**

Learning is a process by which students develop relatively permanent change in mental associations through experience. This is how learning is defined by cognitive psychologists. Behavioral; psychologists define learning as a relatively permanent change in behavior.

There are following domains of learning:

- A: Cognitive Domain relates to intellectual skills or abilities
- B: Affective Domain relates to emotions, feelings, likes, dislikes etc.
- C: Psychomotor Domain relates to manipulative skills of hands, legs. Eye-hand coordination in Engineering & Technology courses, endeavor is made to design curriculum with a focus on development of cognitive skills through classroom teaching. Whereas manipulative (psychomotor) skills are developed in workshops, laboratories & seminars where students work individually or in a group. Development of affective skills attitudes and value is supposed to be acquired through projects and co-curricular activities. These are also developed from the work culture or institutions.

How far a student has developed these abilities/skills especially from cognitive and psychomotor domains is assessed on the basis of suitable examinations. When classroom and laboratory teaching is viewed in this light, evaluation becomes an integral part of teaching – learning process.

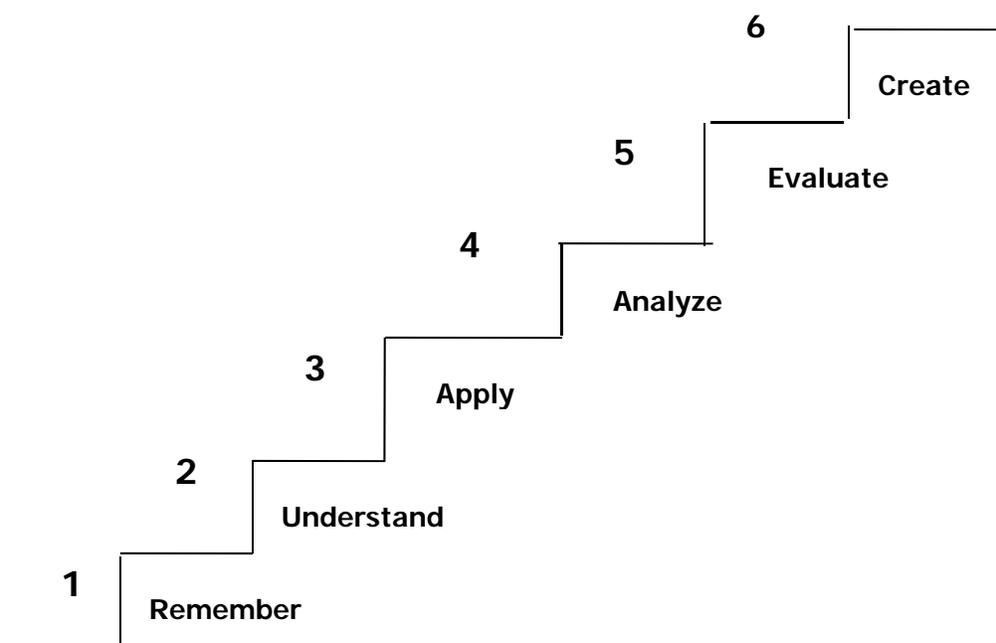
### **2.3 LEVELS OF LEARNING:**

Question paper is a tool/ instrument designed to test the extent of learning of the student. Various questions set in a question paper should assess the abilities of students to respond to level of learning. Dr. Bloom a German educationist classified levels of learning in cognitive domain for the purpose of writing objectives and assessment. Dr. Bloom's revised taxonomy is based on cognitive psychology and is two dimensional. First dimension is cognitive process dimension and other is knowledge dimension. Details of these two dimensions are given below.

#### **2.4.1 Cognitive Domain:**

Dr. Benjamin Bloom (1956) analyzed questions asked in various examinations in American situation and proposed a hierarchical arrangement of instructional objectives (Intellectual abilities) tested by these questions.

The lowest level of cognitive learning achieved by a student is demonstrated by the recall of information that the student retrieves from his long term memory. So, the storage and retrieval of specific facts, concepts, principles, laws, definitions, properties, procedures etc. directly from memory was classified as a knowledge level objective. Thus questions testing memory of students were treated as at the lowest level of the hierarchy of intellectual abilities. The other levels of hierarchy proposed by Dr. Bloom in 1956 relate to the degree of information processing required in the brain needed to provide answer to a question. The various levels in the cognitive hierarchy proposed by Dr. Bloom in 1956 and further revised in 2001 are given below in the diagrammatic form.



Following are the details of each level which indicate the general and specific objectives. Further appropriate verbs are given which are useful in setting good questions. In this table only four levels are considered for diploma students.

Description of the Major Levels in the cognitive Domain (Bloom's Taxonomy)	Illustrative General Instructional Objectives	Illustrative verbs for stating specific learning outcomes
<b>Remember</b> – Knowledge is defined as the remembering of previously learned material. This may involve the recall of a wide range of material, from specific facts to complete theories, but all that is required to mind of the appropriate information. This represents the lowest level of learning outcomes in the cognitive domain	Knows common terms, specific facts, basic concepts, principles, methods & procedures	Define, describe, identify label, list, match, name, outline, reproduce, select, state
<b>Understand</b> – This is defined as the ability to grasp the meaning of material. This may be shown by translating material from one form to another (words or numbers) by interpreting material (explaining or summarizing), and by estimating future trends (predicting consequences or effects). Draw sketches these learning outcomes go one step beyond the simple remembering of material and represent the lowest level of understanding.	Understands fact, principles Interprets verbal material, Interprets charts, tables, graphs. Translates verbal material to mathematical formula. Estimates consequences implied in data. Justifies methods & procedures.	Convert, distinguish estimate, explain, extend, generalize, give examples; infer, paraphrase, predict, rewrite, summarize, draw labeled sketches.

<p><b>Apply</b> – Application refers to the ability to use learned material in new and concrete situations. This may include the application of such things as concepts, principles, rules, methods, laws and theories. Learning outcomes in this area require a higher level of understanding than those under the level described earlier.</p>	<p>Applies principles to new situations. Applies theories to practical situations. Solves mathematical problem. Construct charts, graphs Demonstrates correct usage of a procedure</p>	<p>Change, compile, demonstrate, discover, manipulate, modify, operate, predict, prepare, produce, show, solve, use.</p>
<p><b>Analyze</b> – Analysis refers to the ability to break down material into its component parts so that its organizational structure may be understood. This may include the identification of the parts, analysis of the relationship between parts, and recognition of the organizational principles involved. Learning outcomes here represent a higher intellectual level than “understand” and apply because they require an understanding of both the content and the structural form of the material.</p>	<p>Recognizes unstated assumptions and logical fallacies in reasoning. Distinguishes between facts and inferences. Evaluates relevance/adequacy of data.</p>	<p>Breakdown, diagram, differentiate, discriminate, distinguish, identify, illustrate, infer, outline, point out, relate, select, separate, subdivide.</p>

#### 2.4.2 Categories of Knowledge Dimension

After considering the various designations of knowledge types, especially developments in cognitive psychology that have taken place since the original framework of Bloom’s taxonomy, knowledge is categorized in 4 types – Factual , Conceptual, Procedural and Meta-cognitive.

*Factual Knowledge (A)* is knowledge of discrete, isolated content elements. It includes knowledge of terminology and knowledge of specific details and elements. In contrast, *Conceptual Knowledge (B)* is knowledge of “more complex, organized knowledge form”. It includes knowledge of classifications and categories, principles and generalizations and theories, models and structures.

*Procedural Knowledge (C)* is “knowledge of how to do something”. It includes knowledge of skills and algorithms, techniques and methods, as well as knowledge of criteria used to determine and/or justify “when to do what” within specific fields and disciplines.

*Meta-cognitive knowledge (D)* is “knowledge about cognition in general as well as awareness of and knowledge about one’s own cognition. It encompasses strategic knowledge, knowledge about cognitive tasks, including contextual and conditional knowledge; and self-knowledge”.

Assessment is required to be done on the basis of categories of knowledge and levels of learning. Table below indicates the two dimensional grid based on Blooms Taxonomy for setting questions.

Knowledge Dimension	COGNITIVE PROCESS DIMENSION			
	1 Remember	2 Understand	3 Apply	4 Analyze
A. Factual Knowledge				
B. Conceptual Knowledge				
C. Procedural Knowledge				
D. Meta-cognitive Knowledge				

## 2.5 Components of Curriculum:

**2.5.1 Rationale:** It indicates the logical basis for the inclusion of the subject in the curriculum. It also indicates the importance of the subject related to the entire curriculum.

Rationale tells the students the connection of subjects related to the study of higher level subjects and also the use in their job/profession.

**2.5.2 Objectives:** Objectives indicate what the student will be able to do/perform after he/she completes the study of the subject. It also in other words indicates the scope of the subject.

Objectives indicate what is achievable and hence give direction to the student about how to study the subject, what important things are to be observed and performed during practicals.

Just as rationale indicates the use of the knowledge gained while studying the subject, objectives indicate how efficiently and effectively one can work if the objectives are fulfilled while studying the subject.

**2.5.3 Learning Structure:** It graphically/pictorially indicates the content of the curriculum of the subject and what is to be learnt in the subject. As you know that Cognitive Domain knowledge is divided in four components as mentioned in the Two dimensional grid. Of this Factual, Conceptual and Procedural knowledge components are identified in the curriculum of the subject along with the applications.

Facts, Concepts, Principles are used in developing procedures and applications. So these are given sequentially below procedure as Principles, Concepts and Facts in their order. Learning structure also provide an idea about how to develop the subject logically to achieve the objectives.

**2.5.4 Contents:** List of topics and subtopics to be included in the curriculum of the subject is given in the contents. This helps in achieving the rationale and objectives identified. Contents indicate the importance of the topics, sub topics in development of the subject and accordingly weightages in terms of Hours required to teach the subject components, so that the desired learning takes place. Marks to be allotted while testing the knowledge gained by the student are also indicated.

**2.5.5 Practicals:** While designing the curriculum the objectives are identified. To achieve these objectives students have to develop certain intellectual and motor skills. These skills are developed through well designed Practicals. So in the curriculum the list of the skills to be developed through Practicals is given. The list of Practicals is so developed that after performing the Practicals identified skills will be developed. Here it is necessary that the teacher gives enough opportunity to all the students to perform the practical properly to develop the skills in each one of them.

The skills will be developed if the students actually perform certain activities or tasks. Therefore it is necessary that any practical included in the curriculum necessarily involve some activities to be done by the students. So one has to think and innovate to modify the study experiments so that students will be asked to perform some activity. It could be in terms of identifying components, listing of materials used for manufacturing the components, stating importance of use of certain materials etc.

So any curriculum of a subject is so designed that it achieves the objectives of that subject as well as fulfill the objectives of the entire curriculum

### **3.0 CONTENT ANALYSIS**

#### **3.1 Components of Content Analysis:**

As we have discussed earlier, any curriculum or syllabus of a SUBJECT given to the teacher is organised in terms of UNITS which include TOPICS or SUB-TOPICS as the case may be indicating the TIME in which it is expected to be taught to the students. Components of a topic or part thereof are analysed here at a micro level.

Before we begin actual teaching of any topic (lesson), we must carefully and critically analyse it so that we can plan for teaching - select appropriate media, methods and techniques of teaching and arrange the suitable resources to be required. This analysis

of the content of a Topic results in identification of the following components of the content:

1. Facts
2. Concepts
3. Principles (rules, laws, theories)
4. Applications
5. Procedures
6. Skills (Psychomotor Skills), and
7. Attitudes (underlying affective behaviors as quite often these are not specifically mentioned in the curriculum, still they are to be developed lesson after lesson gradually).

When we undertake the exercise of content analysis, we ourselves understand the subject fully well and at the same time we become clear as to what we are going to teach. It also gives us an idea as to which methods of teaching and media of instruction we should prepare and use and also what resources including time we will require. This analysis will also enable us to design assignments as well as how we are going to assess students learning.

Since the nature of the components of content (1 to 7) differs from one another. These are learned by the students differently as different mental processes are involved in learning these components. The immediate implication of this varying nature of components is that these need to be taught differently and assessed differently. For example, if you look at components 1 to 5 all of which belong to Cognitive Domain of Learning; Component 6 belongs to Psychomotor Domain and Component 7 belongs to Affective Domain (cannot be taught as these attitudes are caught), you will find that these differ from one another. The classification of human behaviors (activities) into the above three domains of learning entails the use of entirely different methods and media of instruction. Different locations of learning (classroom, laboratories, workshops, field visits) need to be selected.

Now we will discuss these components in some detail and see how each one of these should be taught and assessed differently.

### **3.1.1 FACTS:**

These are universally accepted and commonly understood items about which there cannot be much argument and discussion. These are required only to be informed. For example: The sun rises in east and sets in the west; names of scientists and the year in which their theories were propounded; the rules and regulations of admission and examination prescribed by the University are some of the examples of facts. Sometimes, they need not be emphasised in the class as the students already know them. But information can be passed on by word of mouth, if deemed necessary.

### **3.1.2 CONCEPTS:**

A concept is an abstraction or an idea that permits the learner to classify a variety of related phenomena into a convenient and meaningful category. Concept of something is like a picture formation of that thing which helps in conceptualizing it. Gagne says that concept learning produces a certain fundamental change in human performance that is independent of subject or content. Concepts can be divided into the following two categories:

**1. Concrete Concepts:** those which can be seen, touched and manipulated e.g. house, book, table, chair, cat, dog, any machine or apparatus, overhead projector, chalkboard and duster.

**2. Abstract Concepts:** those which cannot be seen and touched and handled but can only be imagined e.g. force, work, fractions, decimal, bending moment, moment of inertia, friction, heat, and induction. Teaching of concrete concepts is not that difficult because the teacher can show the object physically or its picture. On the contrary, teaching of an abstract concept offers difficulty to the teacher as well as for students to understand. These concepts can be learned by heart without understanding as children mug up Nursery Rhymes without understanding even a single word. But at the stage of higher learning, this type of rote learning is not desirable. Adolescents (teenagers) and adults do not accept things without understanding.

### **3.1.3 Concept Attributes:**

We identify a concept and understand it, once we are told about its qualities characteristics, and features. They are technically called concept attributes. While

teaching a concept to our students we must spell out as many attributes as possible for better understanding of the concept.

**Example: The Concept of Friction**

**Attributes:**

1. Friction is a resistive force.
2. Frictional force acts in the direction opposite to the direction of the applied force.
3. Frictional force is more when the surfaces in contact are rough.
4. Smooth surfaces (perfect) have zero friction.
5. Frictional force is self-adjusting to a limit.

Towards the end of this Theme Paper a number of examples of concept attributes are given for your guidance.

The following questions pertaining to a concept (object or process) will be helpful in writing concept attributes:

1. What it is.
2. What are its constituent parts.
3. How it works.
4. How it is similar to and different from other known concepts.
5. What are its uses?

**3.1.4 PRINCIPLES:**

A principle is a statement of relationship between two or more concepts. Principles are sometimes called rules, laws or generalizations. In others words, relationship between two or more concepts which is scientific and universally true is called a Principle.

*For Example:* (related concepts are underlined>)

1. Actions and reactions are equal and opposite.
2. Ohm's law  $I = V/R$  is a principle, where I (Current), V (Voltage), and R (Resistance) are the concepts. While teaching a principle we must recall the concepts which it involves. These concepts might have been taught in the previous lesson. As you already know, concept learning is a prerequisite to Principle learning. Thus we

recall the concepts of current, voltage and resistance by asking questions to the students. Only after that we must tell the relationship among these i.e. Ohm's Law.

### **3.1.5 APPLICATIONS:**

Whatever principles, laws and theories have been learned are only academic exercises unless these are applied to solve a practical problem. In other words, we call this application transfer of learning to a new situation. If you recall, the process of learning dealt with in Theme Paper 2, you will appreciate that the litmus test of learning having occurred is its application in a new situation or solving a new problem.

#### ***For example:***

1. Ohm's law can be applied to find out the unknown quantity (voltage, current, and resistance).
2. Design of a structure can be made based on related principles and theories.
3. Principles of learning and events of instruction can be applied in 'Designing a lesson Plan' and 'Presenting the lesson in the classroom'.
4. The above principles can also be applied while preparing textbooks, workbooks, learning packages and laboratory manuals to be used by the students.

### **3.1.6 PROCEDURES:**

While analysing the content of a topic you might come across certain standard procedures which are prescribed to perform an operation or a given task. These procedures should be clearly identified and taught accordingly not to be left to chance. We should not pre-suppose that the students understand them. We cannot afford to take these things for granted.

#### ***For Example:***

1. Procedure of setting up of an apparatus.
2. Procedure to start an engine.
3. Procedure to operate a machine (a lathe).

### **3.1.7 SKILLS (PSYCHOMOTOR):**

A skill is an ability to perform a task expertly and well. The skilled performance; must meet a pre-specified standard of acceptable performance. A skill has the following three characteristics:

1. It represents a chain of motor responses;

2. It involves the co-ordination of hand and eye movements, and
3. It requires the organization of chains into complex response patterns.

Skills could be intellectual (thinking, understanding); interactive (communication skills) and social (socialising, mixing up with others) also. But normally when we use the word skills, it refers to psychomotor skills.

***For Example:***

1. Welding a butt joint,
2. Setting a theodolite at a station,
3. Making proper circuit connections, and
4. Turning a job on a lathe machine.

Laboratories and workshops of Polytechnics are the locations where these skills are developed among the students under the guidance of expert instructors *of* operators. Drill and practice are the main methods of teaching and learning these skills through model demonstrations and careful observations thereof.

Alongside developing these skills, desirable attitudes like cooperation, team work, leadership, safety, cost consciousness are also developed.

### **3.2 TEACHING OF CONCEPTS;**

In order to teach concepts effectively the following steps have been suggested by De Cecco & Crawford (1974).

**Steps Suggested:**

1. Describe the performance expected of the student after he has learned the concept.
2. Reduce the number of attributes to be learned in complex concepts and make important attributes dominant.
3. Provide the student with verbal indicators (explanation).
4. Provide positive and negative examples (non-examples) of the concept.
5. Present the examples in close succession or simultaneously.
6. Provide occasions for student responses and the reinforcement of these responses, and
7. Assess the learning of the concept.

### **3.3 TEACHING OF PRINCIPLES:**

De Cecco & Crawford (1974) has suggested the following steps for teaching principles effectively.

#### **Steps:**

1. Describe the performance expected of the student after he has learned the principle.
2. Decide and indicate which concepts or principles the students must recall in learning the new principle.
3. Assist the student in the recall of component concepts.
4. Help the student in the recall of component concepts.
5. Help the student to combine the concepts and put them in a proper order.
6. Provide for practice of the principle and for reinforcement of student responses.
7. Assess the learning of the principle.

### **3.4 CONCLUSION:**

To sum up, it can be said that. it is essential for the teachers to develop the skills of 'Content Analysis' of their subjects. It brings content clarity amongst the teachers themselves. More importantly, Content Analysis will be a pre-requisite for writing Instructional Objectives of the topic to be taught. Teaching and learning process is bound to be effective once these crucial academic activities are undertaken.

## 4.0 CURRICULUM

**Course Name : Computer Engineering Group**

**Course Code : CO/CM/CW/IF/CD**

**Semester : Sixth for CO/CM/CW/IF and Seventh for CD**

**Subject Title : Advanced Java Programming**

**Subject Code : 17625**

### Teaching and Examination Scheme

Teaching Scheme			Examination Scheme					
TH	TU	PR	PAPER HRS	TH	PR	OR	TW	TOTAL
03	--	04	02	100#*	50#	--	50@	200

**\*- Online Examination**

#### NOTE:

- **Two tests each of 25 marks to be conducted as per the schedule given by MSBTE.**
- **Total of tests marks for all theory subjects are to be converted out of 50 and to be entered in mark sheet under the head Sessional Work (SW).**

#### Rationale:

Now days, Internet has touched every aspect of life. If we are not connected to internet, it is like we are nowhere. Online presence is a must for businesses. If your enterprise is not online, you are far behind. Web presence has dominated the businesses worldwide.

Java technology is more suitable for web applications development. It has market dominance in the development of online applications. Java is the preferred choice of the programmers and the enterprises globally.

This subject will equip the students with the required knowledge and the skill needed for the development of robust, powerful and scalable enterprise level web applications. It gives students hands-on experience on GUI Technologies viz. AWT and Swings, event handling mechanisms and network programming. Security issues are also taken into considerations.

The most important aspect of web applications - Database Interaction - is also nicely covered. The performance critical areas of the online applications which the Java technology

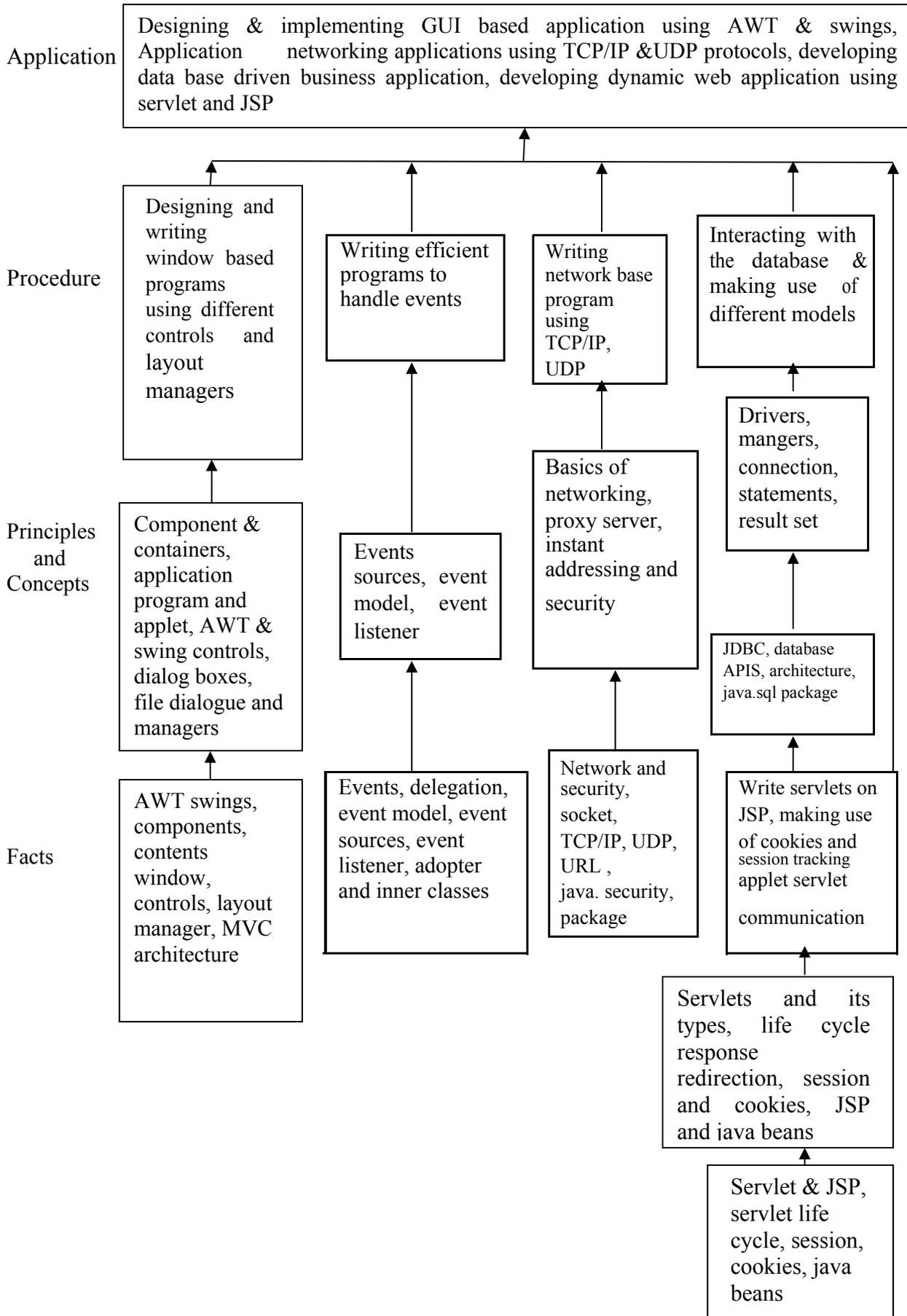
deals with the ease and in a flexible manner by the use of advanced server side components - servlets - are also systematically covered. The students will be able to understand the concepts like servlet chaining, filtering, sessions, cookies and the most important Applet - Servlet communication. Students will also learn the JSP and the Java Beans.

**General Objectives:**

**Intellectual skills:**

- Use of programming language constructs.
- To know apply different logics to solve the given problem.
- To be able to write program using different implementations for the same problem.
- Study different types of errors.
- Debugging of programs.
- Understand different steps to develop program such as
  - a. Problem definition
  - b. Analysis
  - c. Design of logic
  - d. Coding
  - e. Testing
  - f. Maintenance

## Learning Structure:



## Contents:

Topic No.	Name of the Topic	Hours	Marks
01	<p><b>Introduction to Abstract Windowing Toolkit(AWT) &amp; Swings</b></p> <p>Specific Objective</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> <input type="checkbox"/> To design &amp; develop Graphical user interface (GUI) programs using AWT and swing component.</li> <li><input type="checkbox"/> <input type="checkbox"/> To arrange the GUI components using different layout managers.</li> </ul> <p>1.1 Component, container, window, frame, panel.</p> <p>1.2 Creating windowed programs &amp; applets.</p> <p>1.3 AWT controls &amp; layout managers</p> <p>Understanding the use of AWT controls: labels, buttons, checkbox, checkbox group, scroll bars, text field, text area</p> <p>Understanding the use of layout managers: flowLayout, BorderLayout, GridLayout, CardLayout, GridBagLayout, menubars, menus, dialog boxes, file dialog.</p> <p>1.4 Introduction to swing</p> <p>Swing features, MVC Architecture, Combo Boxes, progress bar, tool tips, seperator, tables, trees, toggle button.</p>	16	24
02	<p><b>Event Handling</b></p> <p>Specific Objective</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> <input type="checkbox"/> To write event driven programs using the delegation event model.</li> <li><input type="checkbox"/> <input type="checkbox"/> To write programs using adapter classes &amp; the inner classes.</li> </ul> <p>2.1 The delegation Event Model</p> <p>Event sources, Event listeners, Event classes. The Action Event class, The Component Event class, the Container Event class, the Focus Event class, the Item Event class, the Key Event class, the Mouse Event class, the Text Event class, the Window Event class.</p> <p>2.2 Adapter classes</p> <p>2.3 Inner classes</p> <p>2.4 Event listener interfaces</p> <p>The ActionListener Interface, the ComponentListener Interface, the ContainerListener Interface, the FocusListener Interface, the ItemListener Interface, the KeyListener Interface, the MouseListener Interface, the MouseMotion Interface, the TextListener Interface, the WindowsListener Interface, the WindowFocusListener Interface</p>	10	20

03	<p><b>Networking &amp; Security</b>  Specific Objective:  <input type="checkbox"/> <input type="checkbox"/> To learn the Java's built in support for network programming.  <input type="checkbox"/> <input type="checkbox"/> To write program to demonstrate connectivity through software SOCKETS, TCP, ISP, URL and the Java security package.</p> <p>3.1 Basics of Networking  Socket, IP, TCP, UDP, Proxy Server, Internet Addressing</p> <p>3.2 The InetAddress Class  Factory methods  Instance methods</p> <p>3.3 TCP/IP Sockets  Socket, Server Socket, methods</p> <p>3.4 URL  URL Connection, http, URL Connection methods, creating &amp; using TCP/IP client &amp; server</p> <p>3.5 Security with Java: Theoretical introduction to java.security Package  Permission class  Policy class</p>	08	16
04	<p><b>Interacting with Database</b>  Specific Objective :</p> <p><input type="checkbox"/> <input type="checkbox"/> To create database driven business applications using the database API'S two tier and three tier models and the Java.Sql package</p> <p>4.1 JDBC, ODBC, &amp; Other APIS  JDBC two tier &amp; three tier models</p> <p>4.2 Connecting to Database  Driver Interface, Driver Manager class, Connection Interface, Statement Interface, the java.sql.package  Establishing connection &amp; retrieving information Resultset interface.</p>	06	20
05	<p><b>Servlets &amp; JSP</b>  Specific Objectives :</p> <p><input type="checkbox"/> <input type="checkbox"/> To write web based applications using servlets, JSP and Java Beans.  <input type="checkbox"/> <input type="checkbox"/> To write servlet for cookies and session tracking.</p> <p>5.1 Servlet  Type of Servlet, Servlet life cycle.</p> <p>5.2 Using servlets, response redirection.</p> <p>5.3 Basic concepts of sessions, cookies &amp; session tracking</p> <p>5.4 Introduction to servlet chaining &amp; filters, Introduction to applet servlet communication.</p> <p>5.5 JSP, expression, directives&amp; declarations,  Life cycle of a JSP page TLD &amp; JSTL, Java beans.</p>	08	20
<b>Total</b>		<b>48</b>	<b>100</b>

**List of Practical:**

Sr. No.	Title of Experiment	No. of Hours
1	Write a program to design a form using the components textfield, label, checkbox, button, list.	2
2	Write a program to demonstrate the use of Border layout showing four buttons at four sides of an applet with captions left, right, top and bottom.	2
3	Write a program using AWT to create a menubar in a frame where menubar contains menu items such as File, Edit, View and the submenu under the File menu item should contain New and Open	2
4	Write a program using swing to display a JComboBox in an applet with the items – cricket, football, hockey, tennis	2
5	Write a program to create a Jtree and recognize mouse clicks on it.	4
6	Write a program to create a JTable On JApplet Window.	4
7	Write a program to display the key pressed on Applet Window.	4
8	Write a program to perform addition of two nos. make use of textfield and button.	4
9	Write a program making use of Adapter class.	4
10	Write a program to retrieve hostname and IP Address in InetAddress class.	4
11	Write a program to use URL connection class and display 1) Protocol 2) HostName 3) PortNumber 4) File Name.	4
12	Write a program that demonstrates TCP/IP based communication between Client and Server. Client send “HELLO” to Server and Server replies “HI...” to Client.	4
13	Write a program to send data to Table “XYZ” in database using prepared statement and retrieve data from same Table “XYZ” and display on screen.	4
14	Write a Servlet to display the user name and password accepted from the client.	4
15	Write a Servlet for demonstrating the concept of Session and Cookie.	4
16	Write a simple Program to design a login JSP pages.	4
17	Mini Project	8
<b>Total Hours</b>		<b>64</b>

**Learning Resources:****Books to be referred:**

Sr. No	Author	Title	Publisher
1	Herbert Sheild	Complete Reference	Tata McGraw
2	Kogent learning Solution	Advance JAVA	DreamTech Press
3	Sharnam Shah & Vaishali Shah	Java EE6 for Beginners	SPD
4	Kogent learning Solution	Java Server Programming Black Book	DreamTech Press

**Practical Contents:**

Student will install the following software under the guidance of their Teacher.

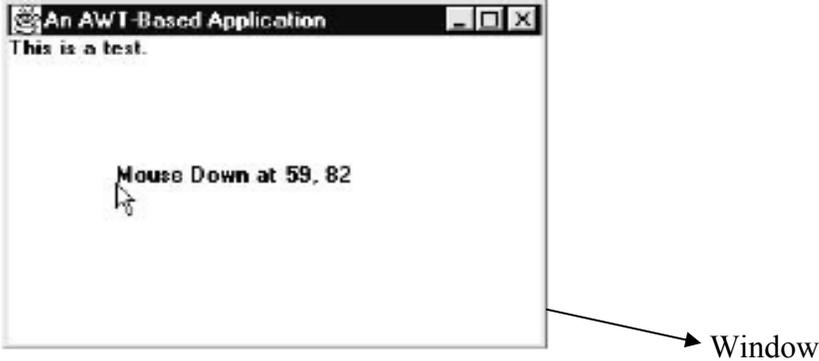
- 1) JDK 1.5 or higher, JRE ( JAVA SOFTWARE)
- 2) NetBeans (or any IDE)
- 3) Database (any one )
- 4) Tomcat web Server
- 5) Special attention on Servlet and JSP from Projects point of view.

## 5. IMPLEMENTATION STRATEGY:

### 5.1 Planning of Lectures for a Semester with Content Detailing :

Topic I	<p><b>Name: Introduction to Abstract Window toolkit (AWT) and Swings</b></p> <p><b>Specific Objectives: (As per Curriculum)</b></p> <ol style="list-style-type: none"> <li>To design and develop Graphical User Interface (GUI) program using AWT and Swing components.</li> <li>To arrange the GUI components using different Layout managers.</li> </ol> <p><b>Additional Specific Objective</b></p> <ul style="list-style-type: none"> <li>To be able to write and execute programs on AWT and Swing.</li> <li>To be able to solve MCQs.</li> </ul>		
	<b>Knowledge Category</b>	<b>Example/s of category</b>	<b>Teaching methodology</b>
	<b>FACT</b>	GUI, AWT and Swing components used in GUI	Examples of different GUI that the students have seen
	<b>CONCEPT</b>	Selection of components for GUI, Layout of different components	Component seen in different applications
	<b>PROCEDURE</b>	Creating objects of different components, Adding the objects to containers, Placement of Components	Programs showing the use of constructors of different components, Adding component object to container.
	<b>APPLICATION</b>	Solving MCQ's for theory examination	Practice of programs in laboratory
	<p>Learning Resources:</p> <p>Books:</p> <ol style="list-style-type: none"> <li>Complete Reference Herbert Schildt 5th Edition Mc Graw Hill</li> <li>Java EE6 for Beginners Sharnam Shah &amp; Vaishali Shah SPD</li> <li>Advanced JAVA Kogent learning Solution DreamTech Press</li> <li>Java server Programming Black Book Kogent learning Solution DreamTech Press</li> </ol> <p>Teaching Aids:</p> <p>Black board, chalk, Transparencies, PowerPoint presentations, References, Books, notes, LCD Projector.</p> <p>PPTs:</p> <ol style="list-style-type: none"> <li><a href="http://www.cpsc.ucalgary.ca/~schock/java/CH15AWTComponents.ppt">www.cpsc.ucalgary.ca/~schock/java/CH15AWTComponents.ppt</a></li> </ol>		

	<p>2. <a href="http://www.sci.brooklyn.cuny.edu/~zhou/teaching/java/awt.ppt">www.sci.brooklyn.cuny.edu/~zhou/teaching/java/awt.ppt</a></p> <p>Websites</p> <ol style="list-style-type: none"> <li>1. <a href="http://www.cs.princeton.edu/courses/archive/spr96/cs333/.../components.html">www.cs.princeton.edu/courses/archive/spr96/cs333/.../components.html</a></li> <li>2. <a href="http://www.tutorialspoint.com/awt/awt_controls.htm">www.tutorialspoint.com/awt/awt_controls.htm</a></li> </ol>
Lecture No.	Topic/ Subtopic to be covered
1	<p>1.1 Component, Container, Window, Frame, Panel</p> <div data-bbox="456 541 1279 919" data-label="Diagram"> <pre> graph TD     Component --&gt; Container     Container --&gt; Window     Container --&gt; Panel     Window --&gt; MenuContainerInterface[MenuContainer Interface]     Window --&gt; Frame   </pre> </div> <p><b>Component:</b> At the top of the AWT hierarchy is the <b>Component</b> class. Component is an abstract class that encapsulates all of the attributes of a visual component. All user interface elements that are displayed on the screen and that interact with the user are subclasses of <b>Component</b>.</p> <p><b>Container:</b> The <b>Container</b> class is a subclass of <b>Component</b>. It has additional methods that allow other <b>Component</b> objects to be nested within it.</p> <p><b>Window:</b> The <b>Window</b> class creates a top-level window. A <i>top-level window</i> is not contained within any other object; it sits directly on the desktop.</p> <p><b>Frame:</b> It is a subclass of <b>Window</b> and has a title bar, menu bar, borders, and resizing corners.</p> <p><b>Panel:</b> A <b>Panel</b> is a window that does not contain a title bar, menu bar, or border.</p> <p><i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 691 to 693)</i></p>
2	1.2 Creating Windowed Program

	 <p>To create such output following concepts are used:</p> <ol style="list-style-type: none"> <li>1. Create a frame window</li> <li>2. Create a window</li> <li>3. Create an object of a component.</li> <li>4. Add the object to frame.</li> </ol> <p><i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 702 to 704)</i></p>
3	<p>1.3 <i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 736 to 763)</i>        AWT Controls and Layout managers        AWT and Swing controls are used to design a form like MSBTE exam form, First year Admission form        Understanding use of AWT controls: Labels, Buttons.        A label is used to display a message to user.</p> <p><b>Label:</b> A <i>label</i> is an object of type <b>Label</b>, and it contains a string, which it displays. Labels are passive controls that do not support any interaction with the user.</p> <p><b>Constructors:</b>        Label()        Label(String <i>str</i>)        Label(String <i>str</i>, int <i>how</i>)</p> <p>A Button is used to take user input when user wants to indicate that GUI form has been filled</p> <p><b>Button:</b> A <i>push button</i> is a component that contains a label and that generates an event when it is pressed.</p> <p><b>Constructors:</b>        Button()        Button(String <i>str</i>)</p>
4	<p>Checkbox, Checkbox group        A <i>check box</i> is used to turn an option on or off.  <b>Checkbox:</b> A <i>check box</i> is a control that is used to turn an option on or</p>

	<p>off. It consists of a small box that can either contain a check mark or not. There is a label associated with each check box that describes what option the box represents.</p> <p><b>Constructor</b>  Checkbox( )  Checkbox(String <i>str</i>)  Checkbox(String <i>str</i>, boolean <i>on</i>)  Checkbox(String <i>str</i>, boolean <i>on</i>, CheckboxGroup <i>cbGroup</i>)  Checkbox(String <i>str</i>, CheckboxGroup <i>cbGroup</i>, boolean <i>on</i>)</p> <p>Checkbox group creates a set of <b>mutually exclusive check boxes</b> in which one and only one check box in the group can be checked at any one time.</p> <p><b>Checkbox group:</b> Creates a set of <b>mutually exclusive check boxes</b> in which one and only one check box in the group can be checked at any one time. These check boxes are often called <i>radio buttons</i>.</p>
5	<p><b>Scroll bar</b>  Scroll bar: The slider box in the scroll bar can be dragged by the user to a new position. The scroll bar will reflect this value.  <i>Scroll bar:</i> Scroll bars are used to select continuous values between a specified minimum and maximum. Scroll bars may be oriented horizontally or vertically.</p> <p><b>Constructors:</b>  Scrollbar( )  Scrollbar(int <i>style</i>)  Scrollbar(int <i>style</i>, int <i>initialValue</i>, int <i>thumbSize</i>, int <i>min</i>, int <i>max</i>)</p>
	<p>Text Field, Text Area.  <b>Text Field is used to take user input when a text such as name, username, password etc is entered in GUI</b>  <b>TextField</b> class implements a single-line text-entry area, usually called an <i>edit control</i>. Text fields allow the user to enter strings and to edit the text using the arrow keys, cut and paste keys, and mouse selections.</p> <p><b>Constructors:</b>  TextField( )  TextField(int <i>numChars</i>)  TextField(String <i>str</i>)  TextField(String <i>str</i>, int <i>numChars</i>)</p> <p><b>TextArea is used to take user input when a text such as address and information which is more than one line is entered in GUI</b></p> <p>TextArea( )  TextArea(int <i>numLines</i>, int <i>numChars</i>)  TextArea(String <i>str</i>)  TextArea(String <i>str</i>, int <i>numLines</i>, int <i>numChars</i>)  TextArea(String <i>str</i>, int <i>numLines</i>, int <i>numChars</i>, int <i>sBars</i>)</p>
7	<p>Understanding the use of layout managers:  (Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 763 to 775)</p>

	<p>flow layout, border layout</p> <p>A layout manager automatically arranges your controls within a window.  <b>FlowLayout:</b> In this Layout components are laid out from the upper-left corner, left to right and top to bottom. When no more components fit on a line, the next one appears on the next line. A small space is left between each component, above and below, as well as left and right.</p> <p><b>Constructors:</b>  FlowLayout( )  FlowLayout(int <i>how</i>)  FlowLayout(int <i>how</i>, int <i>horz</i>, int <i>vert</i>)</p> <p><b>BorderLayout:</b> It has four narrow, fixed-width components at the edges and one large area in the center. The four sides are referred to as north, south, east, and west. The middle area is called the center.</p> <p><b>Constructors:</b>  BorderLayout( )  BorderLayout(int <i>horz</i>, int <i>vert</i>)</p>
8	<p>grid layout, card layout.</p> <p><b>GridLayout:</b> It lays out components in a two-dimensional grid. When you instantiate a <b>GridLayout</b>, you define the number of rows and columns.</p> <p><b>Constructors</b>  GridLayout( )  GridLayout(int <i>numRows</i>, int <i>numColumns</i> )  GridLayout(int <i>numRows</i>, int <i>numColumns</i>, int <i>horz</i>, int <i>vert</i>)</p> <p>CardLayout: The <b>CardLayout</b> class is unique among the other layout managers in that it stores several different layouts. Each layout can be thought of as being on a separate index card in a deck that can be shuffled so that any card is on top at a given time.</p> <p><b>Constructors</b>  CardLayout( )  CardLayout(int <i>horz</i>, int <i>vert</i>)</p>
9	<p><b>grid bag layout</b></p> <p>It aligns components by placing them within a grid of cells, allowing components to span more than one cell. The rows in the grid can have different heights, and grid columns can have different widths.</p> <p>Constructor:  GridBagLayout()</p>
10	<p>Menubars, Menus, Dialog boxes,  <i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 775 to 790)</i></p> <p>A menu bar displays a list of top-level menu choices. Each choice is associated with a drop-down menu. a menu bar contains one or more <b>Menu</b></p>

	<p>objects. Each <b>Menu</b> object contains a list of <b>MenuItem</b> objects. Each <b>MenuItem</b> object represents something that can be selected by the user.</p> <p><b>Constructors for Menu:</b>  Menu( )  Menu(String <i>optionName</i>)  Menu(String <i>optionName</i>, boolean <i>removable</i>)</p> <p><b>Constructors for MenuItem:</b>  MenuItem( )  MenuItem(String <i>itemName</i>)  MenuItem(String <i>itemName</i>, MenuShortcut <i>keyAccel</i>)</p>
11	<p>Dialog Boxes, File dialog  <b>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 775 to 790)</b></p> <p><b>Dialog Box:</b> Dialog boxes are primarily used to obtain user input. They are similar to frame windows, except that dialog boxes are always child windows of a top-level window. Dialog boxes don't have menu bars. Dialog boxes may be modal or modeless. When a <i>modal</i> dialog box is active, all input is directed to it until it is closed. When a <i>modeless</i> dialog box is active, input focus can be directed to another window in your program</p> <p><b>Constructors:</b>  Dialog(Frame <i>parentWindow</i>, boolean <i>mode</i>)  Dialog(Frame <i>parentWindow</i>, String <i>title</i>, boolean <i>mode</i>)</p> <p><b>FileDialog:</b> A built-in dialog box provided by Java that lets the user to specify a file.</p> <p><b>Constructors:</b>  FileDialog(Frame <i>parent</i>, String <i>boxName</i>)  FileDialog(Frame <i>parent</i>, String <i>boxName</i>, int <i>how</i>)  FileDialog(Frame <i>parent</i>)</p>
12	<p>1.4 <b>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 922 to 948)</b>  Introduction to Swing  Swing features, MVC architecture.</p>
13	<p>Combo boxes, progress bar, tool tips  <b>(Java 7 Programming Kogent Learning Solutions Inc)(Page nos. 551 to 561)</b></p> <p><b>ComboBox:</b> Combo boxes can be used as normal combo box as well as drop down list. The user can make selection by clicking on the arrow.</p> <p><b>Constructors.</b>  JComboBox( JComboBoxModel a Model)  JComboBox(Object [] items)  JComboBox(Vector&lt;?&gt; items)</p>

	<p><b>ProgressBar:</b> It draws an updated colored bar that displays the progress of some operation.</p> <p><b>Constructors.</b>  JProgressBar()  JProgressBar(BoundedRangeModel newModel)  JProgressBar(int orient)  JProgressBar(int min, int max)  JProgressBar(int orient, int min, int max )</p> <p><b>Tool tips:</b> These are small windows that appear when the user lets rest on a component.</p> <p><b>Constructor.</b>  JToolTip()</p>
14	<p>separator, tables  <i>(Java 7 Programming Kogent Learning Solutions Inc)(Page nos. 551 to 561)</i></p> <p><b>Separator:</b> These are horizontal or vertical lines and they usually appear in menus to separate options into logical groups.</p> <p><b>Constructor.</b>  JSeparator()  JSeparator(int orientation)</p> <p>Tables: A <i>table</i> is a component that displays rows and columns of data.</p> <p><b>Constructor.</b>  JTable(Object <i>data</i>[ ][ ], Object <i>colHeads</i>[ ])</p>
15	<p>trees, Toggle Button  <i>(Java 7 Programming Kogent Learning Solutions Inc)(Page nos. 551 to 561)</i></p> <p><b>Tree:</b> A <i>tree</i> is a component that presents a hierarchical view of data. A user has the ability to  expand or collapse individual subtrees in this display.</p> <p><b>Constructor.</b>  JTree(Hashtable <i>ht</i>)  JTree(Object <i>obj</i>[ ])  JTree(TreeNode <i>tn</i>)  JTree(Vector <i>v</i>)</p> <p><b>Toggle Button: They are two state buttons that can appear as selected or deleted.</b></p> <p><b>Constructors.</b>  JToggleButton()</p>

	<p>JToggleButton(Action a)  JToggleButton(Icon icon)  JToggleButton(String text)  JToggleButton(String text, boolean selected)  JToggleButton(String text, Icon icon)  JToggleButton(String text, Icon icon, boolean selected)</p>															
16	Revision															
Topic II	<p><b>Name: Event Handling</b>  <b>Specific Objectives (As per curriculum)</b></p> <ol style="list-style-type: none"> <li>To write event driven programs using the delegation event model.</li> <li>To write programs using adapter classes &amp; the inner classes.</li> </ol> <p><b>Additional Specific Objective</b></p> <ul style="list-style-type: none"> <li>To be able to write and execute programs on AWT and Swing using Event Handling.</li> <li>To be able to solve MCQs.</li> </ul> <table border="1"> <thead> <tr> <th>Knowledge Category</th> <th>Example/s of category</th> <th>Teaching methodology</th> </tr> </thead> <tbody> <tr> <td><b>FACT</b></td> <td>Event Source, Event class, Event Listener</td> <td>Event, Examples of Events, Working of Event Classes/Listeners</td> </tr> <tr> <td><b>CONCEPT</b></td> <td>Event Delegation Model</td> <td>Working of Event Delegation Model</td> </tr> <tr> <td><b>PROCEDURE</b></td> <td>Use of Listeners in programs to handle different type of events</td> <td>Student shall write programs based on event handling for different AWT and Swing Components.</td> </tr> <tr> <td><b>APPLICATION</b></td> <td>Solving MCQ's for final theory examination</td> <td>Practice of programs in laboratory</td> </tr> </tbody> </table> <p>Learning Resources:</p> <p>Books:</p> <p>Title: Books:</p> <ol style="list-style-type: none"> <li>Complete Reference Herbert Schildt 5th Edition Mc Graw Hill</li> <li>Java EE6 for Beginners Sharnam Shah &amp; Vaishali Shah SPD</li> <li>Advanced JAVA Kogent learning Solution DreamTech Press</li> <li>Java server Programming Black Book Kogent learning Solution DreamTech Press</li> </ol> <p>Teaching Aids:  Black board, chalk, Transparencies, PowerPoint presentations, References, Books, notes, LCD Projector.</p>	Knowledge Category	Example/s of category	Teaching methodology	<b>FACT</b>	Event Source, Event class, Event Listener	Event, Examples of Events, Working of Event Classes/Listeners	<b>CONCEPT</b>	Event Delegation Model	Working of Event Delegation Model	<b>PROCEDURE</b>	Use of Listeners in programs to handle different type of events	Student shall write programs based on event handling for different AWT and Swing Components.	<b>APPLICATION</b>	Solving MCQ's for final theory examination	Practice of programs in laboratory
Knowledge Category	Example/s of category	Teaching methodology														
<b>FACT</b>	Event Source, Event class, Event Listener	Event, Examples of Events, Working of Event Classes/Listeners														
<b>CONCEPT</b>	Event Delegation Model	Working of Event Delegation Model														
<b>PROCEDURE</b>	Use of Listeners in programs to handle different type of events	Student shall write programs based on event handling for different AWT and Swing Components.														
<b>APPLICATION</b>	Solving MCQ's for final theory examination	Practice of programs in laboratory														

	<p>PPTs:</p> <ol style="list-style-type: none"> <li>1. <a href="http://www.cs.kau.se/cs/education/courses/davd11/.../java_event_handling.ppt">www.cs.kau.se/cs/education/courses/davd11/.../java_event_handling.ppt</a></li> <li>2. <a href="http://web.njit.edu/~gblank/cs602/GUI%20Event%20Handling.ppt">web.njit.edu/~gblank/cs602/GUI%20Event%20Handling.ppt</a></li> </ol> <p>Websites</p> <ol style="list-style-type: none"> <li>1. <a href="http://www.tutorialspoint.com/awt/awt_event_handling.htm">www.tutorialspoint.com/awt/awt_event_handling.htm</a></li> <li>2. <a href="http://www.studytonight.com/java/event-handling-in-java.php">www.studytonight.com/java/event-handling-in-java.php</a></li> </ol>
Lecture No.	Topic/ Subtopic to be covered
1	<p>2.1 (<i>Complete reference Java 2 – 5<sup>th</sup> Edition</i>) (page nos 654 to 668)</p> <p><b>The Delegation Event Model</b></p> <p>A <i>source</i> generates an event and sends it to one or more <i>listeners</i>. In this scheme, the listener simply waits until it receives an event. Once received, the listener processes the event and then returns. The advantage of this design is that the application logic that processes events is cleanly separated from the user interface logic that generates those events.</p> <p><b>Event Sources</b></p> <p>A <i>source</i> is an object that generates an event. This occurs when the internal state of that object changes in some way</p> <p><b>Event Listeners</b></p> <p>A <i>listener</i> is an object that is notified when an event occurs. It has two major requirements. First, it must have been registered with one or more sources to receive notifications about specific types of events. Second, it must implement methods to receive and process these notifications.</p> <p><b>Event classes</b></p> <p>The classes that represent events are at the core of Java’s event handling mechanism.</p> <p><b>The Action Event class</b></p> <p>An <b>ActionEvent</b> is generated when a button is pressed, a list item is double-clicked, or a menu item is selected</p> <p><b>The Component Event class</b></p> <p>A <b>ComponentEvent</b> is generated when the size, position, or visibility of a component is changed</p> <p><b>The Container Event class</b></p> <p>A <b>ContainerEvent</b> is generated when a component is added to or removed from a container.</p>
2	<p><b>The Focus Event class</b></p> <p>A <b>FocusEvent</b> is generated when a component gains or loses input focus</p> <p><b>The Item Event class</b></p> <p>An <b>ItemEvent</b> is generated when a check box or a list item is clicked or when a checkable menu item is selected or deselected</p> <p><b>The Key Event class.</b></p> <p>A <b>KeyEvent</b> is generated when keyboard input occurs</p>

3	<p>The Mouse Event class</p> <p>There are eight types of mouse events. MOUSE_CLICKED, MOUSE_DRAGGED, MOUSE_ENTERED, MOUSE_EXITED, MOUSE_MOVED, MOUSE_PRESSED, MOUSE_RELEASED, MOUSE_WHEEL</p> <p><b>The Text Event class</b></p> <p>These are generated by text fields and text areas when characters are entered by a user or program</p> <p><b>The Window Event class.</b></p> <p>There are ten types of window events.  WINDOW_ACTIVATED, WINDOW_CLOSED, WINDOW_CLOSING ,  WINDOW_DEACTIVATED, WINDOW_DEICONIFIED,  WINDOW_GAINED_FOCUS, WINDOW_ICONIFIED  WINDOW_LOST_FOCUS, WINDOW_OPENED WINDOW_STATE_CHANGED</p>
4	<p>2.4 (<i>Complete reference Java 2 – 5<sup>th</sup> Edition</i>) (page nos 668 to 679)</p> <p><b>The Event Listener Interfaces</b></p> <p>The delegation event model has two parts: sources and listeners. Listeners are created by implementing one or more of the interfaces defined by the <b>java.awt.event</b> package. When an event occurs, the event source invokes the appropriate method defined by the listener and provides an event object as its argument.</p> <p><b>The ActionListener Interface</b></p> <p>This interface defines the <b>actionPerformed()</b> method that is invoked when an action event occurs. Its general form is shown here:  void actionPerformed(ActionEvent ae)</p> <p><b>The ComponentListener Interface</b></p> <p>This interface defines four methods that are invoked when a component is resized, moved, shown, or hidden. Their general forms are shown here:  void componentResized(ComponentEvent ce)  void componentMoved(ComponentEvent ce)  void componentShown(ComponentEvent ce)  void componentHidden(ComponentEvent ce)</p> <p><b>The ContainerListener Interface</b></p> <p>This interface contains two methods. When a component is added to a container, <b>componentAdded()</b> is invoked. When a component is removed from a container, <b>componentRemoved()</b> is invoked. Their general forms are shown here:  void componentAdded(ContainerEvent ce)  void componentRemoved(ContainerEvent ce)</p>
5	<p><b>The FocusListener Interface</b></p> <p>This interface defines two methods. When a component obtains keyboard focus, <b>focusGained()</b> is invoked. When a component loses keyboard focus, <b>focusLost()</b> is called. Their general forms are shown here:  void focusGained(FocusEvent fe)  void focusLost(FocusEvent fe)</p> <p><b>The ItemListener Interface</b></p> <p>This interface defines the <b>itemStateChanged()</b> method that is invoked when the state of an item changes. Its general form is shown here:  void itemStateChanged(ItemEvent ie)</p>

6	<p><b>The KeyListener Interface</b>  This interface defines three methods. The <b>keyPressed()</b> and <b>keyReleased()</b> methods are invoked when a key is pressed and released, respectively. The <b>keyTyped()</b> method is invoked when a character has been entered.</p> <pre>void keyPressed(KeyEvent ke) void keyReleased(KeyEvent ke) void keyTyped(KeyEvent ke)</pre> <p><b>The MouseListener Interface</b>  This interface defines five methods. If the mouse is pressed and released at the same point, <b>mouseClicked()</b> is invoked. When the mouse enters a component, the <b>mouseEntered()</b> method is called. When it leaves, <b>mouseExited()</b> is called. The <b>mousePressed()</b> and <b>mouseReleased()</b> methods are invoked when the mouse is pressed and released, respectively.  The general forms of these methods <pre>void mouseClicked(MouseEvent me) void mouseEntered(MouseEvent me) void mouseExited(MouseEvent me) void mousePressed(MouseEvent me) void mouseReleased(MouseEvent me)</pre> </p>
7	<p><b>The MouseMotionListener Interface</b>  This interface defines two methods. The <b>mouseDragged()</b> method is called multiple times as the mouse is dragged. The <b>mouseMoved()</b> method is called multiple times as the mouse is moved. Their general forms are shown here:</p> <pre>void mouseDragged(MouseEvent me) void mouseMoved(MouseEvent me)</pre> <p><b>The TextListener Interface</b>  This interface defines the <b>textChanged()</b> method that is invoked when a change occurs in a text area or text field. Its general form is shown here:</p> <pre>void textChanged(TextEvent te)</pre>
8	<p><b>The WindowListener Interface</b>  This interface defines seven methods. The <b>windowActivated()</b> and <b>windowDeactivated()</b> methods are invoked when a window is activated or deactivated, respectively. If a window is iconified, the <b>windowIconified()</b> method is called. When a window is deiconified, the <b>windowDeiconified()</b> method is called. When a window is opened or closed, the <b>windowOpened()</b> or <b>windowClosed()</b> methods are called, respectively. The <b>windowClosing()</b> method is called when a window is being closed. The general forms of these methods are</p> <pre>void windowActivated(WindowEvent we) void windowClosed(WindowEvent we) void windowClosing(WindowEvent we) void windowDeactivated(WindowEvent we) void windowDeiconified(WindowEvent we) void windowIconified(WindowEvent we) void windowOpened(WindowEvent we)</pre> <p><b>The WindowFocusListener Interface.</b>  This interface defines two methods: <b>windowGainedFocus()</b> and <b>windowLostFocus()</b>. These are called when a window gains or losses input</p>

	<p>focus. Their general forms are shown here.</p> <pre>void windowGainedFocus(WindowEvent we) void windowLostFocus(WindowEvent we)</pre>												
9	<p><b>2.2 The Adapter classes</b>  <i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 680 to 681)</i></p> <p><i>Adapter class</i> can simplify the creation of event handlers in certain situations. An adapter class provides an empty implementation of all methods in an event listener interface. Adapter classes are useful when you want to receive and process only some of the events that are handled by a particular event listener interface.</p>												
10	<p><b>2.3 The Inner classes</b>  <i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 682 to 685)</i></p> <p><i>Inner class</i> is a class defined within other class, or even within an expression.</p>												
	<p><b>After covering AWT components, a teacher may cover the portion of event handling from second chapter. Remaining portion of chapter 1 may be covered after event handling so that students may get more exposure to application and remember level questions</b></p>												
Topic III	<p><b>Name: Networking &amp; Security</b>  <b>Specific Objectives: (As per Curriculum)</b></p> <ol style="list-style-type: none"> <li>To learn the java's built in support for network programming</li> <li>To write program to demonstrate the connectivity through software sockets, TCP, IP, URL and java security packages.</li> </ol> <p><b>Additional Specific Objectives</b></p> <ul style="list-style-type: none"> <li>To be able to write networking programs</li> <li>To be able to use java security package.</li> <li>To be able to solve MCQs</li> </ul> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Knowledge Category</th> <th style="width: 25%;">Example/s of category</th> <th style="width: 50%;">Teaching methodology</th> </tr> </thead> <tbody> <tr> <td><b>FACT</b></td> <td><b>URL, Socket, InetAddress class</b></td> <td>Use following PPT  <a href="http://www.cse.unr.edu/~mgunes/cpe401/cpe401sp09/Lecture11.ppt">www.cse.unr.edu/~mgunes/cpe401/cpe401sp09/Lecture11.ppt</a></td> </tr> <tr> <td><b>CONCEPT</b></td> <td><b>TCP, UDP, Proxy Server</b></td> <td>Use of following PPT  <a href="http://www.ircbeginner.com/ircinfo/Routing_Article.pdf">http://www.ircbeginner.com/ircinfo/Routing_Article.pdf</a></td> </tr> <tr> <td><b>PROCEDURE</b></td> <td><b>URL Connection, Creating and using TCP/IP Client &amp; Server</b></td> <td>Use of following PPT  <a href="http://www.cafeaulait.org/slides/sd99east/URLS.PPT">www.cafeaulait.org/slides/sd99east/URLS.PPT</a></td> </tr> </tbody> </table>	Knowledge Category	Example/s of category	Teaching methodology	<b>FACT</b>	<b>URL, Socket, InetAddress class</b>	Use following PPT <a href="http://www.cse.unr.edu/~mgunes/cpe401/cpe401sp09/Lecture11.ppt">www.cse.unr.edu/~mgunes/cpe401/cpe401sp09/Lecture11.ppt</a>	<b>CONCEPT</b>	<b>TCP, UDP, Proxy Server</b>	Use of following PPT <a href="http://www.ircbeginner.com/ircinfo/Routing_Article.pdf">http://www.ircbeginner.com/ircinfo/Routing_Article.pdf</a>	<b>PROCEDURE</b>	<b>URL Connection, Creating and using TCP/IP Client &amp; Server</b>	Use of following PPT <a href="http://www.cafeaulait.org/slides/sd99east/URLS.PPT">www.cafeaulait.org/slides/sd99east/URLS.PPT</a>
Knowledge Category	Example/s of category	Teaching methodology											
<b>FACT</b>	<b>URL, Socket, InetAddress class</b>	Use following PPT <a href="http://www.cse.unr.edu/~mgunes/cpe401/cpe401sp09/Lecture11.ppt">www.cse.unr.edu/~mgunes/cpe401/cpe401sp09/Lecture11.ppt</a>											
<b>CONCEPT</b>	<b>TCP, UDP, Proxy Server</b>	Use of following PPT <a href="http://www.ircbeginner.com/ircinfo/Routing_Article.pdf">http://www.ircbeginner.com/ircinfo/Routing_Article.pdf</a>											
<b>PROCEDURE</b>	<b>URL Connection, Creating and using TCP/IP Client &amp; Server</b>	Use of following PPT <a href="http://www.cafeaulait.org/slides/sd99east/URLS.PPT">www.cafeaulait.org/slides/sd99east/URLS.PPT</a>											

	<b>APPLICATION</b>	<b>Chat Application</b>	Use of appropriate PPT
Lecture No.	<b>Topic/ Subtopic to be covered</b>		
1	<p> <b>Basics of Networking</b>          Socket, IP, TCP, UDP, Proxy Server, Internet Addressing          Socket Overview, Client/Server ,Reserved Sockets ,Proxy Servers ,Internet Addressing  <b>Socket:</b> A socket can be used to connect Java's I/O system to other programs that may reside either on the local machine or on any other machine on the Internet.  <b>TCP:</b> Transmission Control Protocol (TCP) is a higher-level protocol that manages to robustly string together these packets, sorting and retransmitting them as necessary to reliably transmit your data.  <b>UDP :</b>A third protocol, User Datagram Protocol (UDP), sits next to TCP and can be used directly to support fast, connectionless, unreliable transport of packets.  <b>Proxy Server:</b> A proxy server speaks the client side of a protocol to another server. This is often required when clients have certain restrictions on which servers they can connect to.          Thus, a client would connect to a proxy server, which did not have such restrictions, and the proxy server would in turn communicate for the client. A proxy server has the additional ability to filter certain requests or cache the results of those requests for future use.       </p> <p> <b>Internet Addressing:</b> Every computer on the Internet has an address. An Internet address is a number that uniquely identifies each computer on the Net.  <i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 588 to 591)</i>  <a href="http://www.sfml-dev.org/tutorials/1.6/network-sockets.php">http://www.sfml-dev.org/tutorials/1.6/network-sockets.php</a> </p>		

	<a href="http://www.cse.unr.edu/~mgunes/cpe401/cpe401sp09/Lecture11.ppt">www.cse.unr.edu/~mgunes/cpe401/cpe401sp09/Lecture11.ppt</a>
2	<p><b>The InetAddress Class</b></p> <p><b>Factory methods:</b> Factory methods are merely a convention whereby static methods in a class return an instance of that class. This is done in lieu of overloading a constructor with various parameter lists when having unique method names makes the results much clearer. Three commonly used InetAddress factory methods are:</p> <pre>static InetAddress getLocalHost( )     throws UnknownHostException static InetAddress getByName(String hostName)     throws UnknownHostException static InetAddress[ ] getAllByName(String hostName)     throws UnknownHostException</pre> <p><i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 592 to 593)</i></p> <p><a href="http://docs.oracle.com/javase/7/docs/api/java/net/class-use/InetAddress.html">http://docs.oracle.com/javase/7/docs/api/java/net/class-use/InetAddress.html</a></p>
3	<p><b>Instance methods:</b> Some commonly used methods are:</p> <pre>boolean equals(Object other) byte[ ] getAddress( ) String getHostAddress( ) String getHostName( ) String toString( )</pre> <p><i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 594)</i></p> <p><a href="http://docs.oracle.com/javase/7/docs/api/java/net/class-use/InetAddress.html">http://docs.oracle.com/javase/7/docs/api/java/net/class-use/InetAddress.html</a></p>
4	<p><b>TCP/IP Sockets</b></p> <p>TCP/IP sockets are used to implement reliable, bidirectional, persistent, point-to-point, stream-based connections between hosts on the Internet.</p> <p><b>Server Socket</b></p> <p>The ServerSocket class is designed to be a “listener,” which waits for clients to connect before doing anything. The Socket class is designed to connect to server sockets and initiate protocol exchanges.</p> <p><a href="http://www.cs.rutgers.edu/~pxk/rutgers/notes/sockets/">http://www.cs.rutgers.edu/~pxk/rutgers/notes/sockets/</a></p> <p><b>TCP/IP Sockets Methods</b></p> <p>A socket can be examined at any time for the address and port information associated with it, by use of the following methods:</p> <pre>InetAddress getInetAddress( ) Returns the InetAddress associated with the Socket object. int getPort( ) Returns the remote port to which this Socket object is connected. int getLocalPort( ) Returns the local port to which this Socket object is connected.</pre> <p><i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 594 to 595)</i></p> <p><a href="http://www.tutorialspoint.com/java/java_networking.htm">http://www.tutorialspoint.com/java/java_networking.htm</a></p>

5	<p><b>URL:</b> The URL provides a reasonably intelligible form to uniquely identify or address information on the Internet. URLs are ubiquitous; every browser uses them to identify information on the Web. Two examples of URLs are <code>http://www.osborne.com/</code> and <code>http://www.osborne.com:80/index.htm</code>.</p> <p><b>URL Connection:</b> URL Connection is a general-purpose class for accessing the attributes of a remote resource. Once you make a connection to a remote server, you can use URL Connection to inspect the properties of the remote object before actually transporting it locally. These attributes are exposed by the HTTP protocol specification and, as such, only make sense for URL objects that are using the HTTP protocol.</p> <p><b>http:</b>http is the protocol that web browsers and servers use to transfer hypertext pages and images. It is quite a simple protocol for a basic page-browsing web server.</p> <p><b>URLConnection methods:</b> <b>URLConnection</b> is a general-purpose class for accessing the attributes of a remote resource. Once you make a connection to a remote server, you can use URLConnection to inspect the properties of the remote object before actually transporting it locally.</p> <p><b>openConnection()</b> (<i>Complete reference Java 2 – 5<sup>th</sup> Edition</i>) (page nos 597 to 601)</p> <p><a href="http://www.cse.unr.edu/~mgunes/cpe401/cpe401sp09/Lecture11.ppt">www.cse.unr.edu/~mgunes/cpe401/cpe401sp09/Lecture11.ppt</a></p>
6	<p><b>Creating &amp; Using TCP/IP client &amp; server</b> TCP/IP sockets are used to implement reliable, bidirectional, persistent, point-to-point, stream-based connections between hosts on the Internet.</p> <p><b>TCP/IP Client Sockets:</b> There are two kinds of TCP sockets in Java. One is for servers, and the other is for clients. The ServerSocket class is designed to be a “listener,” which waits for clients to connect before doing anything. The Socket class is designed to connect to server sockets and initiate protocol exchanges. Here are two constructors used to create client sockets: Socket(String hostName, int port): Creates a socket connecting the local host to the named host and port; can throw an UnknownHostException or an IOException. Socket(InetAddress ipAddress, int port): Creates a socket using a preexisting InetAddress object and a port; can throw an IOException. pg 594-595 Herbert Sheild Complete Reference Tata McGraw Fifth Edition</p> <p><b>TCP/IP Server Sockets:</b> The ServerSocket class is used to create servers that listen for either local or remote client programs to connect to them on published ports. Since the Web is driving most of the activity on the Internet, this section develops an operational web (http) server. ServerSocket(int port): Creates server socket on the specified port with a queue</p>

	<p>length of 50.</p> <p>ServerSocket(int port, int maxQueue): Creates a server socket on the specified port with a maximum queue length of maxQueue.</p> <p>ServerSocket(int port, int maxQueue, InetAddress localAddress): Creates a server socket on the specified port with a maximum queue length of maxQueue. On a multihomed host, localAddress specifies the IP address to which this socket binds.</p> <p><i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 601 to 602)</i></p> <p><a href="http://www.bogotobogo.com/Java/tutorial/tcp_socket_server_client.php">http://www.bogotobogo.com/Java/tutorial/tcp_socket_server_client.php</a></p>									
7	<p><b>Security with Java: Theoretical introduction to java.security Package</b></p> <p><b>java.security Package :</b>The java.security package contains the classes and interfaces that implement the Java security architecture. These classes can be divided into two broad categories.</p> <p>The two most important classes of the java.security packages are the java.security.Permission and java.security.Policy.</p> <p><b>Permission class:</b> Permission class is an abstract class in the java.security package which represents access to a system resource. A Permission has a name and abstract functions defining the semantics of a particular Permission subclass.</p> <p><b>Policy class:</b> An object of the Policy class is used to find if the code executing in Java runtime environment has permission to perform security-sensitive operations.</p> <p><i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 848 to 852)</i></p> <p><a href="http://docstore.mik.ua/oreilly/java-ent/jnut/ch17_01.htm">http://docstore.mik.ua/oreilly/java-ent/jnut/ch17_01.htm</a></p>									
8	Revision									
<b>Topic IV</b>	<p><b>Name: Interacting with Database</b></p> <p><b>Specific objectives(as mentioned in curriculum)</b></p> <ol style="list-style-type: none"> <li>To create database driven business applications using the database API'S two tier and three tier models and the Java.sql package.</li> </ol> <p><b>Additional objectives</b></p> <ul style="list-style-type: none"> <li>Able to write and execute the program.</li> </ul>									
	<table border="1"> <thead> <tr> <th>Knowledge Category</th> <th>Example(s) of Category</th> <th>Teaching Methodology</th> </tr> </thead> <tbody> <tr> <td><b>FACT</b></td> <td>Jdbc-Odbc</td> <td>Need of Jdbc and Odbc</td> </tr> <tr> <td><b>CONCEPT</b></td> <td>Jdbc,Java API,java.sql package Two-Tier&amp; Three-Tier Models</td> <td>Block diagram of Two-Tier and Three Tier Architecture. Application of Two-Tier and Three-Tier Architecture. Advantages and disadvantages of two tier and Three-Tier Architecture.</td> </tr> </tbody> </table>	Knowledge Category	Example(s) of Category	Teaching Methodology	<b>FACT</b>	Jdbc-Odbc	Need of Jdbc and Odbc	<b>CONCEPT</b>	Jdbc,Java API,java.sql package Two-Tier& Three-Tier Models	Block diagram of Two-Tier and Three Tier Architecture. Application of Two-Tier and Three-Tier Architecture. Advantages and disadvantages of two tier and Three-Tier Architecture.
Knowledge Category	Example(s) of Category	Teaching Methodology								
<b>FACT</b>	Jdbc-Odbc	Need of Jdbc and Odbc								
<b>CONCEPT</b>	Jdbc,Java API,java.sql package Two-Tier& Three-Tier Models	Block diagram of Two-Tier and Three Tier Architecture. Application of Two-Tier and Three-Tier Architecture. Advantages and disadvantages of two tier and Three-Tier Architecture.								

		Types of drivers	Methods of ResultSet, Statement interface and prepared statement interface All types of drivers. <a href="http://www.javatpoint.com/PreparedStatement-interface">http://www.javatpoint.com/PreparedStatement-interface</a> <a href="http://www.tutorialspoint.com/jdbc/jdbc-statements.htm">http://www.tutorialspoint.com/jdbc/jdbc-statements.htm</a>
	<b>PROCEDURE</b>	Stepwise procedure to write code and execute the java program	Write the program using statement and prepared statement interface in classroom
	<b>APPLICATION</b>	e.g. Railway Reservation System, banking System etc.	Useful in client server and web Application. Practice the MCQ's in Laboratory. Practice the program in Laboratory.
<p><b>Reference Material:</b> Books: - Title 1) Advanced Java ,Kogent learning Solution,Dream The Press 2)Java EE6 for Beginners , Sharnam Shah &amp;Vaishali Shah, SPD 3) Java Server Programming Black Book, Kogent learning Solution, DreamTech Press</p> <p><b>Teaching Aids:</b> Black board, Chalk, Transparencies, Power point presentation slides(PPTs), Reference books, notes, LCD projector/OHP Projector,</p> <p><b>Websites:</b> ➤ <a href="http://docs.oracle.com/javase/tutorial/jdbc/overview/">http://docs.oracle.com/javase/tutorial/jdbc/overview/</a></p>			
Lecture No.	<b>Topic/ Subtopic to be covered</b>		
1	<p><b>4.1 JDBC,ODBC and Others APIS</b></p> <ul style="list-style-type: none"> <li>➤ <b>JDBC</b> : It is Java API “Application programming interface” used for connecting programs that is written in Java to the data in a relational database.</li> <li>➤ <b>ODBC:</b> <ul style="list-style-type: none"> <li>▪ It access the DataBase using SQL requests.</li> <li>▪ ODBC converts these SQL requests into a request that the DataBase can understand.</li> </ul> </li> <li>➤ <b>Jdbc API</b> <ul style="list-style-type: none"> <li>▪ The JDBC API is a Java API that can access any kind of tabular data, especially data stored in a Relational Database.</li> <li>▪ JDBC is used to write Java applications that manage these three</li> </ul> </li> </ul>		

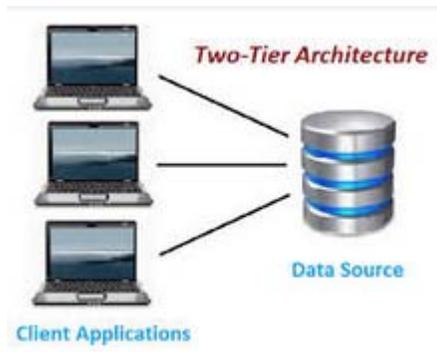
programming activities:

- Connect to a data source, like a database
- Send queries and update statements to the database
- Retrieve and process the results received from the database in answer to your query

(Java 7 Programming Kogent Learning Solutions Inc)(Page nos. 972 - 974)

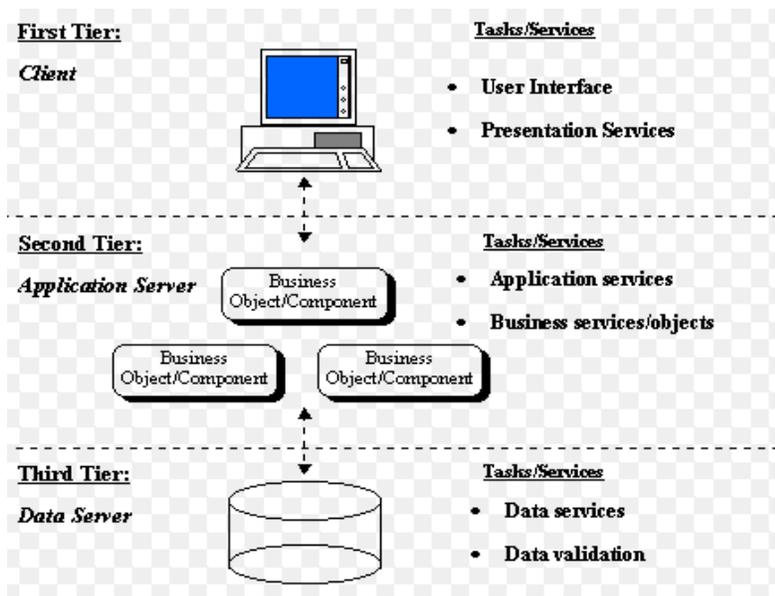
2

### Two -Tier Architecture



- The two-tier architecture is like client server application.
- The direct communication takes place between client and server.
- There is no intermediate between client and server.

### Three- tier Architecture



Three tier architecture having three layers. They are

- Client layer
- Business layer
- Data layer

**Client layer:** Here we design the form using textbox, label etc.

**Business layer:**

- It is the intermediate layer which has the functions for client layer
- It is used to make communication faster between client and data layer.
- It provides the business processes logic and the data access.

**Data layer:** It has the database.

**Advantages two tier architecture:**

- Easy to maintain and modification is bit easy
- Communication is faster

**Disadvantages two tier Architecture:**

- In two tier architecture application performance will be degrade upon increasing the users.
- Cost-ineffective

**Advantages of three tier Architecture**

- Managing data is independent from the physical storage.  
Improve Data Integrity

When one tier fails there is no data loss, because you are always secure by accessing the other tier

**Disadvantages of three tier Architecture**

- It is more complex structure
- More difficult to set up and maintain it as well

*(Java 7 Programming Kogent Learning Solutions Inc)(Page nos. 987)*

#### **4.2 Connecting to Database**

Interfaces and classes required to write the java program

- **DriverMnager class:**-Load driver for database
- **Driver Interface:**-Represents a database driver. All JDBC driver classes must implement the Driver interface
- **Statement interface:** Enable to execute SQL Statements
- **Connection interface:** Establish a connection between a java application and a database
- **ResultSet interface:** Represents the information retrieved from a database
- **SQLException class:** Provides information about the exceptions that occur while interacting with databases

*(Java 7 Programming Kogent Learning Solutions Inc)(Page nos. 989 - 993)*

	<p><b>Type1(Jdbc-Odbc Bridge driver)</b></p> <ul style="list-style-type: none"> <li>▪ Jdbc-Odbc Bridge driver converts JDBC call to ODBC calls .</li> <li>▪ ODBC is an open standard API to communicate with database.</li> <li>▪ It enables a java application to use any database that supports ODBC driver .</li> <li>▪ It is an interface between java application and the ODBC driver.</li> <li>▪ It is usually used in stand-alone application.</li> </ul> <p>➤ <b>Type2(Java to Native API)</b></p> <ul style="list-style-type: none"> <li>▪ It uses the local native libraries provided by the database vendors to access databases.</li> <li>▪ It does not have an ODBC intermediate layer.</li> <li>▪ it is used for network-based application.</li> </ul> <p>➤ <b>Type3(Java to network protocol)</b></p> <ul style="list-style-type: none"> <li>▪ It translates the JDBC calls into a database-server-independent and middleware server-specific calls.</li> <li>▪ With the help of middleware server, the translated JDBC calls are further translated into database server specific calls.</li> </ul> <p>➤ <b>Type4(Java to Database Protocol)</b></p> <ul style="list-style-type: none"> <li>▪ It is pure java driver, which implements the database protocol to interact directly with a database.</li> <li>▪ It does not require any native database library to retrieve the records from the database.</li> <li>▪ . It translates JDBC calls into database specific network calls.</li> </ul> <p><i>(Java 7 Programming Kogent Learning Solutions Inc)(Page nos. 982 - 985)</i></p>
4	<p>➤ <b>Procedure to connect java to MS_ACCESS or any database</b></p> <ul style="list-style-type: none"> <li>▪ Import JDBC packages.</li> <li>▪ Load and register the JDBC driver.</li> <li>▪ Open a connection to the database.</li> <li>▪ Create a statement object to perform a query.</li> <li>▪ Execute the statement object and return a query resultset.</li> <li>▪ Process the resultset.</li> <li>▪ Close the resultset and statement objects.</li> <li>▪ Close the connection.</li> </ul> <p>➤ <b>Statement interface</b></p> <p>Create a <b>Statement</b> object to send request to and retrieve results from a database.</p> <p><b>Connection</b> objects provides the createStatement() method to create a Statement</p> <p>The Statement interface contains the following methods to send static SQL Statements to a database</p>

➤ **Syntax:**intexecuteUpdate(String str);  
 e..g.  
 Statement st=con.createStatement();  
 String str="insert into table1 values(11,'xyz')";  
 int count=st.executeUpdate(str);

➤ **Explain various methods of ResultSet interface**

Methods	Description
boolean first()	Shift the control of a result set cursor to the first row of the result set
booleanisFirst()	Determines whether the result set cursor points to the first row of the result set
Boolean beforeFirst()	Shift the control of a result set cursor before the first row of the result set
booleanisBeforeFirst()	Determines whether the result set cursor points before the first row of the result set
Boolean last()	Shift the control of a result set cursor to the last row of the result set.

➤ **Syntax:** ResultSetexecuteQuery(String str);  
 Statement st=con.createStatement();  
 String str="select \* from table1 ";  
 ResultSetsrs=st.executeQuery(str);

➤ **Syntax:**booleanexecute(String str)  
 Statement st=con.createStatement();  
 String str="create table product (prod\_idint , prod\_name char)";  
 st.execute(str);

➤ **Program based on statement interface such as**

- Create table
- Insert data into table
- Update particular row in table
- Delete row from table

*(Java 7 Programming Kogent Learning Solutions Inc)(Page nos. 1003)*

5

▪ retrieve data from table  
 ➤ **prepared Statement Interface**  
 PreparedStatement interface is derived from Statement interface and is available in java.sql package.

	<p>PreparedStatement object allows to pass runtime parameters to the SQL Statements to query and modify the data in a table</p> <p>The prepareStatement() method of <b>Connection</b> object take an SQL Statement as a parameter. The sql statement can contain ‘?’ symbol as a place holder that can be replaced by input parameters at runtime.</p> <p>➤ <b>Methods of prepared statement</b></p> <ul style="list-style-type: none"> <li>▪ intexecuteUpdate();</li> <li>▪ ResultsetexecuteQuery();</li> <li>▪ Boolean execute()</li> </ul> <table border="1" data-bbox="310 506 1435 1058"> <thead> <tr> <th data-bbox="310 506 829 541">Methods</th> <th data-bbox="829 506 1435 541">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="310 541 829 653">Void setByte(int index, byte value)</td> <td data-bbox="829 541 1435 653">Set the java byte type value for the parameter corresponding to index passed as parameter.</td> </tr> <tr> <td data-bbox="310 653 829 764">Void setBoolean(int index, boolean value)</td> <td data-bbox="829 653 1435 764">Set the java Boolean type value for the parameter corresponding to index passed as parameter.</td> </tr> <tr> <td data-bbox="310 764 829 875">Void setDouble(int index, double value)</td> <td data-bbox="829 764 1435 875">Set the java double type value for the parameter corresponding to index passed as parameter.</td> </tr> <tr> <td data-bbox="310 875 829 947">Void setInt(int index, int value)</td> <td data-bbox="829 875 1435 947">Set the java int type value for the parameter corresponding to index passed as parameter.</td> </tr> <tr> <td data-bbox="310 947 829 1058">Void setString(int index, string value)</td> <td data-bbox="829 947 1435 1058">Set the java string type value for the parameter corresponding to index passed as parameter.</td> </tr> </tbody> </table> <p><i>(Java 7 Programming Kogent Learning Solutions Inc)(Page nos. 1002, 1015)</i></p>	Methods	Description	Void setByte(int index, byte value)	Set the java byte type value for the parameter corresponding to index passed as parameter.	Void setBoolean(int index, boolean value)	Set the java Boolean type value for the parameter corresponding to index passed as parameter.	Void setDouble(int index, double value)	Set the java double type value for the parameter corresponding to index passed as parameter.	Void setInt(int index, int value)	Set the java int type value for the parameter corresponding to index passed as parameter.	Void setString(int index, string value)	Set the java string type value for the parameter corresponding to index passed as parameter.
Methods	Description												
Void setByte(int index, byte value)	Set the java byte type value for the parameter corresponding to index passed as parameter.												
Void setBoolean(int index, boolean value)	Set the java Boolean type value for the parameter corresponding to index passed as parameter.												
Void setDouble(int index, double value)	Set the java double type value for the parameter corresponding to index passed as parameter.												
Void setInt(int index, int value)	Set the java int type value for the parameter corresponding to index passed as parameter.												
Void setString(int index, string value)	Set the java string type value for the parameter corresponding to index passed as parameter.												
6	<p>➤ Program based on prepared statement interface such as</p> <ul style="list-style-type: none"> <li>➤ Create table</li> <li>➤ Insert data into table</li> <li>➤ Update particular row in table</li> <li>➤ Delete row from table</li> <li>➤ Retrieve data from table</li> </ul> <p><i>(Java 7 Programming Kogent Learning Solutions Inc)(Page nos. 1033)</i></p>												
Topic V	<p><b>Name: Servlets and JSP</b></p> <p><b>Specific Objectives (as mentioned in curriculum)</b></p> <ol style="list-style-type: none"> <li>1. To write web based applications using servlets, JSP and Java Beans.</li> <li>2. To learn about cookies session tracking.</li> </ol> <p><b>Additional Specific Objectives:</b></p> <ul style="list-style-type: none"> <li>• To design web based application</li> <li>• To be able to write and execute programs on Servlet and JSP</li> <li>• To solve MCQs</li> </ul>												

Knowledge Category	Example /s of category	Teaching methodology
<b>FACT</b>	<b>Definition of Servlet</b>	Use of appropriate PPT <a href="http://www.calstatela.edu/faculty/areed2/CIS561/PPTs/Chapter5Slides.ppt">www.calstatela.edu/faculty/areed2/CIS561/PPTs/Chapter5Slides.ppt</a>
<b>CONCEPT</b>	<b>Types of Servlet</b>	Use of appropriate PPT <a href="http://www.calstatela.edu/faculty/areed2/CIS561/PPTs/Chapter5Slides.ppt">www.calstatela.edu/faculty/areed2/CIS561/PPTs/Chapter5Slides.ppt</a>
<b>PROCEDURE</b>	<b>Life Cycle of Servlets, Steps for execution of Program</b>	Use of appropriate PPT; <a href="http://www.calstatela.edu/faculty/areed2/CIS561/PPTs/Chapter5Slides.ppt">www.calstatela.edu/faculty/areed2/CIS561/PPTs/Chapter5Slides.ppt</a> Refer manual experiment number 14 for steps for execution of program
<b>APPLICATION</b>	<b>Designing websites like flipkart, amazon for online shopping, online cricket</b>	Show working of websites like <a href="http://www.flipkart.com">www.flipkart.com</a> , online cricket score, share market.

Learning Resources:

Books:

Title:

1. Complete Reference Herbert Schildt 5th Edition Mc Graw Hill
2. Java EE6 for Beginners Sharnam Shah & Vaishali Shah SPD
3. Advanced JAVA Kogent learning Solution DreamTech Press
4. Java server Programming Black Book Kogent learning Solution DreamTech Press

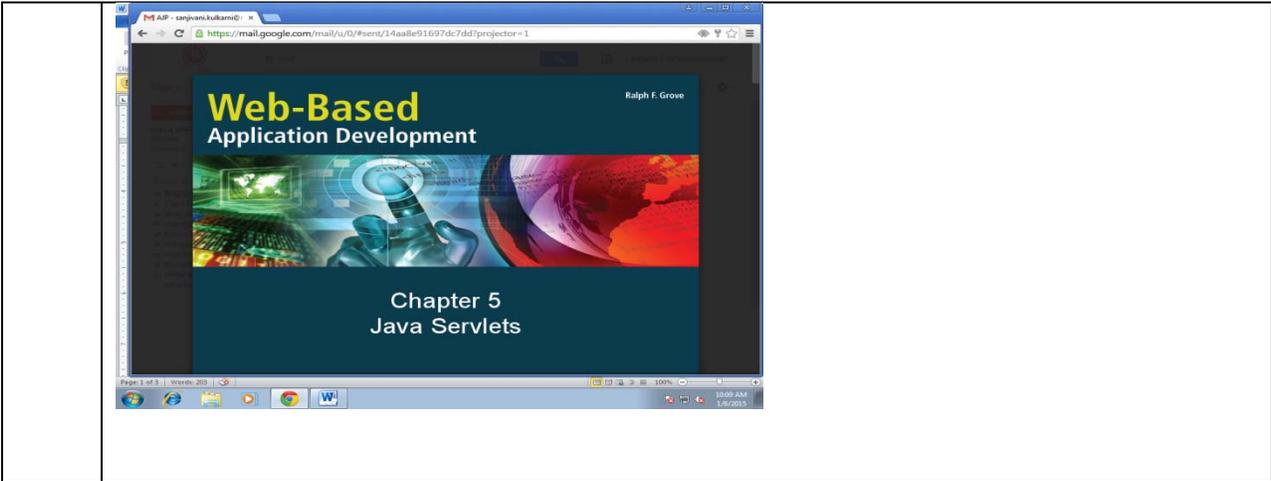
Teaching Aids:

Black board, chalk, Transparencies, PowerPoint presentations, References, Books, notes, LCD Projector.

**PPT with Sample: -**

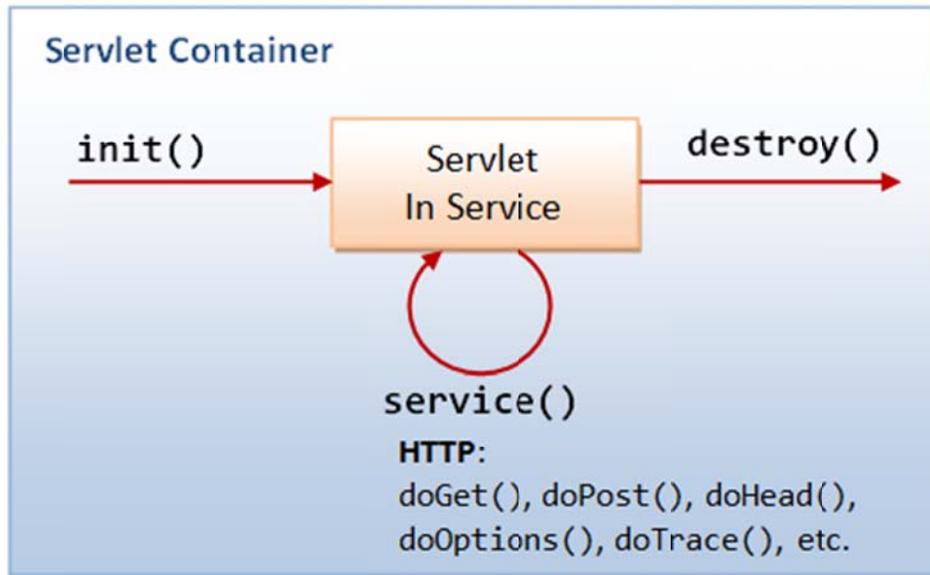
[www.calstatela.edu/faculty/areed2/CIS561/PPTs/Chapter5Slides.ppt](http://www.calstatela.edu/faculty/areed2/CIS561/PPTs/Chapter5Slides.ppt)

(Keyword in Google search:- “Servlet and its types **ppt**” Select “1<sup>st</sup>Link”)



Lecture No.	Topic/ Subtopic to be covered
-------------	-------------------------------

1	<p><b>Servlets &amp; JSP</b></p> <p>5.1 Servlet</p> <p>Everyone things of purchasing of some material from online shopping sites. Designing such site can be done with the help of servlets. So servlets are useful for designing web based applications. Type of Servlets:</p> <p>In the early days of the Web, a server could dynamically construct a page by creating a separate process to handle each client request. The process would open connections to one or more databases in order to obtain the necessary information. It communicated with the Web server via an interface known as the Common Gateway Interface (CGI).</p> <p>CGI allowed the separate process to read data from the HTTP request and write data to the HTTP response. A variety of different languages were used to build CGI programs.</p> <p>These included C, C++, and Perl.</p> <p>Servlets offer several advantages in comparison with CGI. First, performance is significantly better. Servlets execute within the address space of a Web server. It is not necessary to create a separate process to handle each client request. Second, servlets are platform-independent because they are written in Java. A number of Web servers from different vendors offer the Servlet API. Programs developed for this API can be moved to any of these environments without recompilation. Third, the Java security manager on the server enforces a set of restrictions to protect the resources on a server machine. You will see that some servlets are trusted and others are untrusted.</p> <p>Servlet life cycle.</p>
---	--



*(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 951)*

2

## 5.2 Using servlets, response redirection.

The javax.servlet Package

The javax.servlet package contains a number of interfaces and classes that establish the framework in which servlets operate. The following table summarizes the core interfaces that are provided in this package. The most significant of these is Servlet. All servlets must implement this interface or extend a class that implements the interface.

The ServletRequest and ServletResponse interfaces are also very important.

### Interface Description

Servlet Declares life cycle methods for a servlet.

ServletConfig Allows servlets to get initialization parameters.

ServletContext Enables servlets to log events and access information about their environment.

ServletRequest Used to read data from a client request.

ServletResponse Used to write data to a client response.

SingleThreadModel Indicates that the servlet is thread safe.

The following table summarizes the core classes that are provided in the javax.servlet package.

### Class Description

GenericServlet Implements the Servlet and ServletConfig interfaces.

ServletInputStream Provides an input stream for reading requests from a client.

ServletOutputStream Provides an output stream for writing responses to a client.

ServletException Indicates a servlet error occurred.

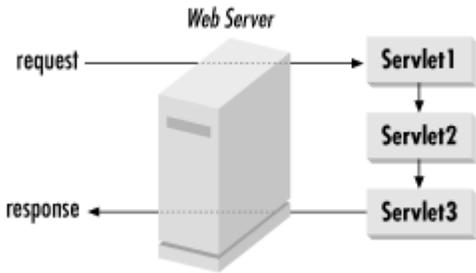
UnavailableException Indicates a servlet is unavailable.

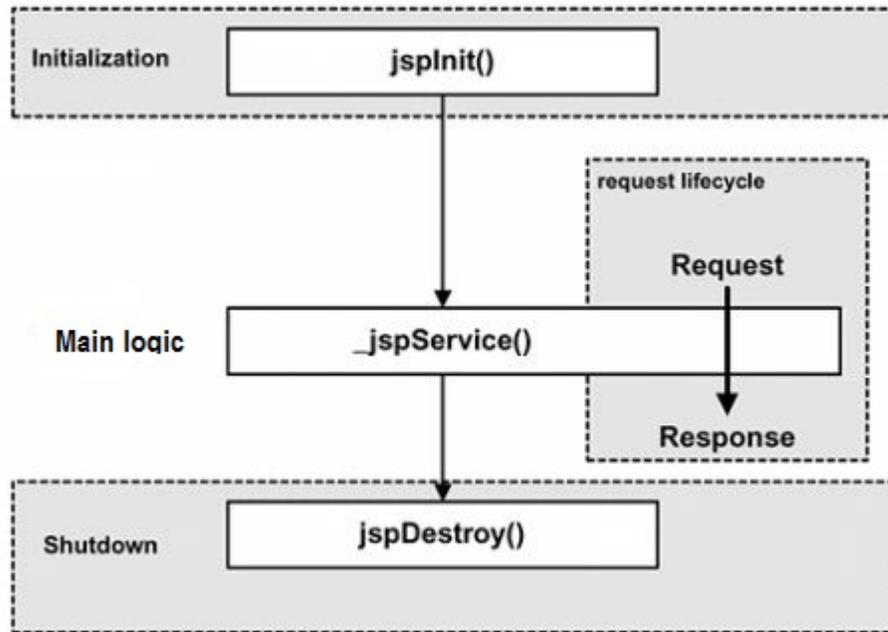
Examine these interfaces and classes in more detail.

3

Show the execution of simple servlet program.

	<i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 961 to 962)</i>
4	<p><b>5.3 Basic concepts of sessions, cookies</b></p> <p><b>Session:</b>  After completion of work with mail account, user of that account performs sign out. So time taken by an account holder from sign in to sign out is called as session.  Describe HttpSessionEvent Class and HttpSessionBindingEvent Class and there methods  Pg nos. 971</p> <p><b>Cookies:</b>  A cookie is stored on a client and contains state information. Cookies are valuable for tracking user activities. For example, assume that a user visits an online store. A cookie can save the user's name, address, and other information. The user does not need to enter this data each time he or she visits the store. A servlet can write a cookie to a user's machine via the addCookie( ) method of the HttpServletResponse interface. The data for that cookie is then included in the header of the HTTP response that is sent to the browser.  <i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 976 to 978)</i></p> <p>Show the execution of Cookies.</p>
5	<p><b>Session tracking:</b></p> <p>Administrator of a system can keep tracking of users on any computer. Person can keep record of who has used the computer. This task can be achieved using concept of session tracking.</p> <p>HTTP is a stateless protocol. Each request is independent of the previous one. However, in some applications, it is necessary to save state information so that information can be collected from several interactions between a browser and a server. Sessions provides such a mechanism.  <i>(Complete reference Java 2 – 5<sup>th</sup> Edition) (page nos 977, 978)</i></p>
6	<b>5.4 Introduction to servlet chaining &amp; filters, Introduction to applet servlet communication.</b>

	<div style="text-align: center;">  <pre> graph TD     Request[request] --&gt; WS[Web Server]     WS --&gt; S1[Servlet1]     S1 --&gt; S2[Servlet2]     S2 --&gt; S3[Servlet3]     S3 --&gt; WS     WS --&gt; Response[response] </pre> </div> <p><a href="http://docstore.mik.ua/oreilly/java-ent/servlet/ch02_05.htm">http://docstore.mik.ua/oreilly/java-ent/servlet/ch02_05.htm</a></p>
7	<p>5.5 JSP, expression, directives&amp; declarations</p> <p>JSP provides far more power than ordinary HTML pages. JSP can also be used as modular, reusable presentation components.</p> <p><a href="http://books.google.co.in/books?id=YUXyUmGpAh8C&amp;pg=PA383&amp;source=gbs_toc_r&amp;cad=2#v=onepage&amp;q&amp;f=false">http://books.google.co.in/books?id=YUXyUmGpAh8C&amp;pg=PA383&amp;source=gbs_toc_r&amp;cad=2#v=onepage&amp;q&amp;f=false</a></p>
8	<p><b>Life cycle of a JSP page TLD &amp; JSTL, Java beans</b></p> <p>JSP life cycle can be defined as the entire process from its creation till the destruction which is similar to a servlet life cycle with an additional step which is required to compile a JSP into servlet.</p> <p>The following are the paths followed by a JSP</p> <ul style="list-style-type: none"> <li>Compilation</li> <li>Initialization</li> <li>Execution</li> <li>Cleanup</li> </ul> <p>The four major phases of JSP life cycle are very similar to Servlet Life Cycle and they are as follows:</p>



### JSP Compilation:

When a browser asks for a JSP, the JSP engine first checks to see whether it needs to compile the page. If the page has never been compiled, or if the JSP has been modified since it was last compiled, the JSP engine compiles the page.

The compilation process involves three steps:

Parsing the JSP.

Turning the JSP into a servlet.

Compiling the servlet.

### JSP Initialization:

When a container loads a JSP it invokes the `jspInit()` method before servicing any requests. If you need to perform JSP-specific initialization, override the `jspInit()` method:

```

public void jspInit() {
    // Initialization code...
}
  
```

Typically initialization is performed only once and as with the servlet `init` method, you generally initialize database connections, open files, and create lookup tables in the `jspInit` method.

### JSP Execution:

This phase of the JSP life cycle represents all interactions with requests until the JSP is destroyed.

Whenever a browser requests a JSP and the page has been loaded and initialized, the JSP engine invokes the `_jspService()` method in the JSP.

The `_jspService()` method takes an `HttpServletRequest` and an `HttpServletResponse` as its parameters as follows:

```

void _jspService(HttpServletRequest request,
    HttpServletResponse response)
  
```

```
{  
// Service handling code...  
}
```

The `_jspService()` method of a JSP is invoked once per a request and is responsible for generating the response for that request and this method is also responsible for generating responses to all seven of the HTTP methods ie. GET, POST, DELETE etc.

### JSP Cleanup:

The destruction phase of the JSP life cycle represents when a JSP is being removed from use by a container.

The `jspDestroy()` method is the JSP equivalent of the destroy method for servlets. Override `jspDestroy` when you need to perform any cleanup, such as releasing database connections or closing open files.

The `jspDestroy()` method has the following form:

```
publicvoidjspDestroy()  
{  
// Your cleanup code goes here.  
}
```

Please see:[http://www.tutorialspoint.com/jsp/jsp\\_life\\_cycle.htm](http://www.tutorialspoint.com/jsp/jsp_life_cycle.htm)

### JAVABeans:( Only introduction)

JavaBeans is an object-oriented programming interface from Sun Microsystems that lets you build re-useable applications or program building blocks called components that can be deployed in a network on any major operating system platform.

This is an example of a simple JavaBean type with the properties `int age` and `String color`.



#### Code listing 9.12: Puppy.java

```
1. class Puppy implements java.io.Serializable{  
2.  
3.  
4.  
5. private static final long serialVersionUID= 348652158488L;  
6.  
7.  
8.  
9. private String color;  
10.  
11.  
12.  
13. private int age;  
14.  
15.  
16.  
17. public String getColor(){  
18.  
19. return color;
```

```

20.
21. }
22.
23.
24.
25. public void setColor(String color){
26.
27.     this.color= color;
28.
29. }
30.
31.
32.
33. public int getAge(){
34.
35.     return age;
36.
37. }
38.
39.
40.
41. public void setAge(int age){
42.
43.     this.age= age;
44.
45. }
46. }

```

## 5.2 Planning and Conduct of Test:

- a) The time table and sample test paper for the test should be displayed minimum 10 days before the test.
- b) Each test will be of 25 marks.
- c) First test should cover about 40% of curriculum and second test should cover remaining curriculum.
- d) Format for question paper should be as per the sample question paper supplied by MSBTE.

### Guidelines for Setting Class Test Question Paper:

- Remember Level 7 Questions 1M each
- Apply/Understand Level 9 Questions 2M each

## 5.3 Details about conduct of assignments:

- After completion of each chapter one assignment in the form of MCQs should be given.
- Assignment question shall be given from sample question paper,.
- It shall be assessed by subject teacher before giving next Assignment.

- Evaluation of Assignment should be done effectively.
- Sample question paper of Advanced Java Programming to be solved by every student

#### 5.4 Strategies for Conduct of Practical:

##### 5.4.1 Suggestions for effective conduct of practical and assessment:

- At the beginning of the semester teacher/lab assistant should check and ensure that the Computers and JDK software used for the experiments is installed.
- Before start of any practical Teachers should explain the specific objective of that particular practical.
- Teacher should divide total students into number of group as given in practical manual.
- Teacher should refer the guidelines given in the lab manual.
- Teacher should make the students aware of instructions given in the lab manual.
- Teacher should ensure that the activities given in the Lab Manual are performed by the student.
- Teacher shall assess the performance of students continuously as per norms prescribed by MSBTE CIAAN norms.
- During assessment teacher is expected to ask questions to the students to tap their achievements regarding related knowledge and skills so that students can prepare while submitting record of the practical. Focus should be given on development of enlisted skills rather than theoretical / codified knowledge.

##### 5.4.3 Preparation for conduct of practical

<b>Expt. No.</b>	<b>Activity</b>	<b>Duration</b>
<b>1.</b>	<b>Write a program to design a form using the components textfield, label, checkbox, button, list.</b>	4 Hrs.
	Teacher shall explain Java Environment with JDK and JRE .	30 Min.
	Teacher shall tell about different IDEs available for Java Programming.	30 Min
	Teacher shall demonstrate installation of JDK software and ask them to install.	45 Min.
	Teacher shall explain basic structure of applet/frame and demonstrate execution of a small program.	15 Min.
	Student should observe options of creating and adding objects of different components.	30 Min.
	Group of two students will perform one allotted activity from student Activity and observe the output.	45 Min.
	Students should write answers to 3 questions given by the teacher.	15 Min.
	Teacher should assess the experiment as per CPA table	30 Min.
<b>2.</b>	<b>Write a program to demonstrate the use of Border</b>	4 Hrs.

	<b>layout showing four buttons at four sides of an applet with captions “left”, “right”, “top” and “bottom”.</b>	
	Teacher shall explain working of Layout Manager and types of layouts	30 Min.
	Teacher shall demonstrate sample program on board.	30 Min.
	Students should type and execute sample program on pc.	20 Min.
	Group of two students should perform one allotted programs from student activity.	40 Min.
	Student should observe the output and get assessed by teacher.	30 Min.
	Students should write observations after execution of programs and take the programs and attach in manuals.	30 Min.
	Students should write answers to 3 questions allotted by teacher.	20 Min.
	Teacher should assess the experiment as per CPA table	40 Min.

## 6. Mode of assessment:

### 6.1.1 Class Test:

- There will be two tests each of 25 marks.
- The tests will be conducted as per the MSBTE schedule.
- Teacher should prepare model answer of class test question papers.
- After completion of test, subject teacher should display model answer on Department Notice Board.
- Teacher should show the answer paper of class test to the student and discuss about the mistakes.
- Teacher should maintain the record of class test as per MSBTE norms (CIAAN)

**Scheme – G**  
**Sample Question Paper Unit Test 1**

**Course Name: - Computer Engineering Group**  
**Course Code:- CO/CD/CM/CW/IF**  
**Semester: - Sixth**  
**Subject Title: - Advanced Java Programming**  
**Marks: - 25 Marks**

**17625**

**Q 1. Which of the following methods can be used to remove java.awt.Component object from display?**

**1M**

- A. hide()
- B. disappear()
- C. remove()
- D. delete()

**Q2. What are controls or components?**

**1M**

- A. Controls or components allow users to interact with application
- B. Controls or components do not allow users to interact with users
- C. Controls or components allow users to interact with users
- D. Controls or components allow application to interact with user

**Q3. What are the subclasses of the Container class?**

**1M**

- A. Windows, Panel, ScrollPane
- B. ScrollPane, Vector, String
- C. Thread, Vector, String
- D. Applet, InetAddress, Vector

**Q4. Which object is needed to group Checkboxes to make them exclusive?**

**1M**

- A. CheckboxGroup
- B. Checkbox
- C. RadioButton
- D. TextField

**Q5. What is an event in delegation event model used by Java programming language? 1M**

- a) An event is an object that describes a state change in a source.
- b) An event is an object that describes a state change in processing.
- c) An event is an object that describes any change by the user and system.

d) An event is a class used for defining object, to create events.

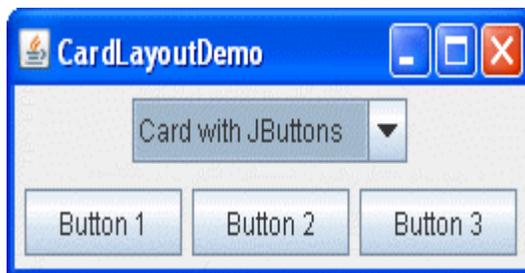
**Q6. Which of these methods are used to register a mouse motion listener? 1M**

- a) addMouse()
- b) addMouseListener()
- c) addMouseMotionListner()
- d) eventMouseMotionListener()

**Q.7 Which of these methods can be used to determine the type of event? 1M**

- a) getID()
- b) getSource()
- c) getEvent()
- d) getEventObject()

**Q8.**



**Which components are needed to get above shown output**

**2M**

- A. TextField, Label
- B. List, Button
- C. Choice, Button
- D. Button, TextField

**Q9. What components will be needed to get following output? 2M**



- A. Label, TabbedPane, CheckBox
- B. TabbedPane, List, Applet
- C. Panel, TabbedPane, List
- D. Applet, TabbedPane, Panel

**Q10. Select the missing statement in given code**

**2M**

```
// Demonstrate the mouse event handlers.
import java.awt.*;
import java.applet.*;
/*
<applet code="mouse" width=300 height=100>
</applet>
*/

public class mouse extends Applet
implements MouseListener, MouseMotionListener
{
String msg = "";
intmouseX = 0, mouseY = 0; // coordinates of mouse
public void init() {
}
// Handle mouse clicked.
public void mouseClicked(MouseEvent me)
{

mouseX = 0;
mouseY = 10;
msg = "Mouse clicked.";
repaint();
}
// Handle mouse entered.
public void mouseEntered(MouseEvent me)
{

mouseX = 0;
mouseY = 10;
msg = "Mouse entered.";
repaint();
}
// Handle mouse exited.
public void mouseExited(MouseEvent me)
{

mouseX = 0;
mouseY = 10;
msg = "Mouse exited.";
repaint();
}
// Handle button pressed.
```

```

public void mousePressed(MouseEvent me)
{

mouseX = me.getX();
mouseY = me.getY();
msg = "Down";
repaint();
}
// Handle button released.
public void mouseReleased(MouseEvent me)
{

mouseX = me.getX();
mouseY = me.getY();
msg = "Up";
repaint();
}
// Handle mouse dragged.
public void mouseDragged(MouseEvent me)
{

mouseX = me.getX();
mouseY = me.getY();
msg = "*";
setStatus("Dragging mouse at " + mouseX + ", " + mouseY);
repaint();
}
// Handle mouse moved.
public void mouseMoved(MouseEvent me)
{
setStatus("Moving mouse at " + me.getX() + ", " + me.getY());
}
// Display msg in applet window at current X,Y location.
public void paint(Graphics g)
{
g.drawString(msg, mouseX, mouseY);
}
}

```

a)addMouseListener(this);

b)addMouseListener(this);  
addMouseMotionListener(this);  
import java.awt.event.\*;

c) addMouseListener();

d) all of above

**Q11. Which of these events will be notified if scroll bar is manipulated?**

**2M**

- a) ActionEvent
- b) ComponentEvent
- c) AdjustmentEvent
- d) WindowEvent

**Q12. Select output for given code**

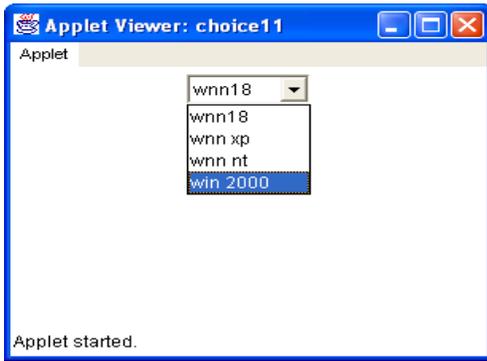
**2M**

```
import java.awt.event.*;
import java.awt.*;
import java.applet.*;
public class checkbackg extends Applet implements ItemListener
{
    Checkbox m1,m2,m3;
    public void init()
    {
        m1=new Checkbox("A");
        m2=new Checkbox("B");
        m3=new Checkbox("C");
        add(m1);
        add(m2);
        add(m3);
        m1.addItemListener(this);
        m2.addItemListener(this);
    }

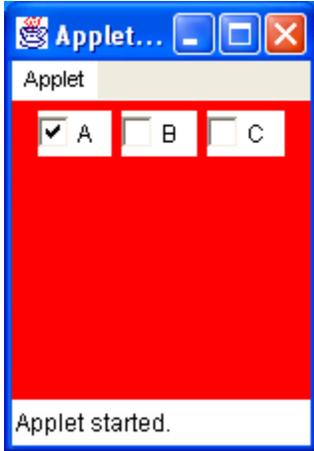
    public void itemStateChanged(ItemEventie)
    {
        if(ie.getSource()==m1)
            setBackground(Color.red);
        if(ie.getSource()==m2)
            setBackground(Color.green);
    }
}
/*<applet code=checkbackg.class height=150 width=150>

</applet>*/
```

a)



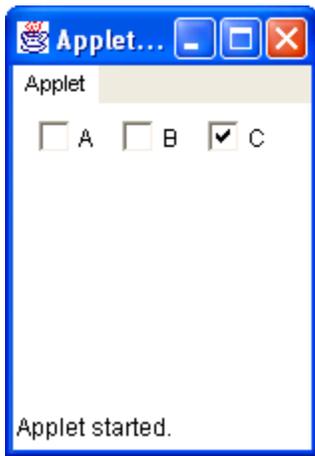
b)



c)

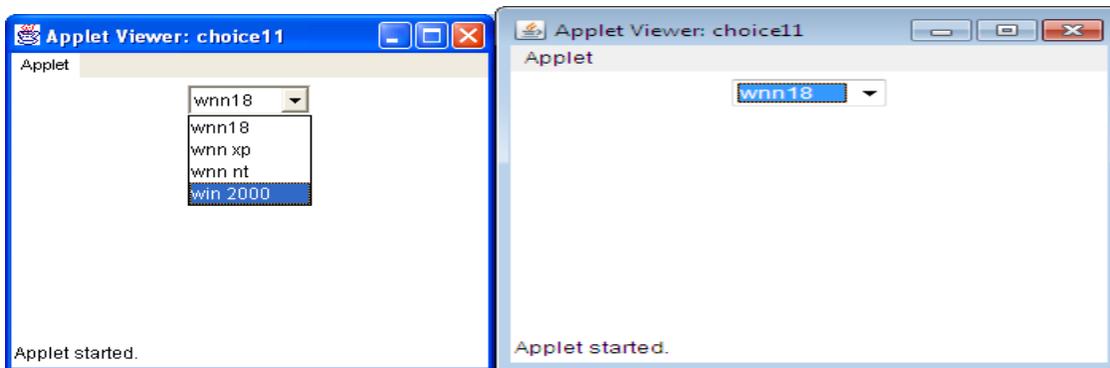


d)



Q13. Select proper code for given output

2M



a) `import java.awt.*;`  
`import java.applet.*;`  
`public class choice11 extends Applet`  
`{`  
`public void init()`  
`{`  
`Choice os=new Choice();`  
`os.add("wnn18");`  
`os.add("wnn xp");`  
`os.add("wnn nt");`  
`os.add("win 2000");`  
`add(os);`  
`}`  
`}`  
`/*<applet code="choice11" height=200 width=300>`  
`</applet>*/`

b)

`import java.awt.*;`  
`import java.applet.*;`  
`public class choice11 extends Applet`

```

{
public void init()
{
Choice os=new Choice();
os.add("wnn18");
os.add("wnnxp");
add(os);
}
}
/*<applet code="choice11" height=200 width=300>
</applet>*/

```

```

c) import java.awt.*;
import java.applet.*;
public class choice11 extends Applet
{
public void init()
{
Choice os=new Choice();
os.add("wnn18");
os.add("wnnxp");
os.add("wnnnt");
os.add("win 2000");
add(os);
}
}

```

```

d)
import java.awt.*;
import java.applet.*;
public class choice11 extends Applet
{

```

```

public void init()
{
Choice os=new Choice();
os.add("wnn18");
os.add("wnnxp");
os.add("wnnnt");
os.add("win 2000");
}
}
/*<applet code="choice11" height=200 width=300>
</applet>*/

```

**Q14. select the proper output for following code**

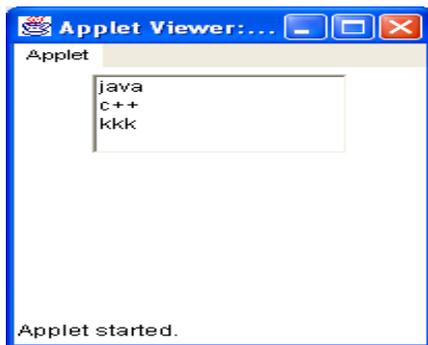
**2M**

```
import java.awt.*;
import java.applet.*;
public class list2 extends Applet
{
public void init()
{
List l= new List(2,true);
l.add("java");
l.add("c++");
l.add("kkk");
add(l);
}
}
/*<applet code=list2.class height=200 width=200>
</applet>*/
```

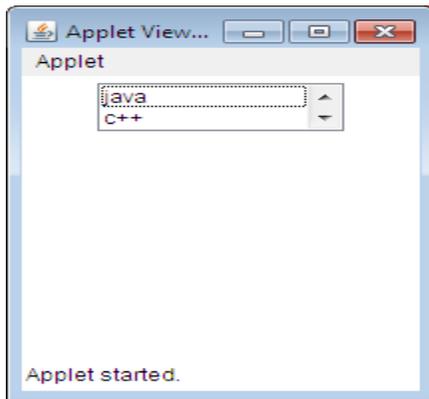
a)



b)



c)



d)



**Q15. Debug the following program**

**2M**

```
import java.awt.*;
import javax.swing.*;
/*
<applet code="JTableDemo" width=400 height=200>
</applet>
*/
public class JTableDemo extends JApplet
{
public void init() {
Container contentPane = getContentPane();
contentPane.setLayout(new BorderLayout());
final String[] colHeads = { "emp_Name", "emp_id", "emp_salary" };
final Object[][] data = {
{ "Ramesh", "111", "50000" },
{ "Sagar", "222", "52000" },
{ "Virag", "333", "40000" },
{ "Amit", "444", "62000" },
{ "Anil", "555", "60000" },
};
}
```

```

};
JTable table = new JTable(data);
int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
JScrollPane jsp = new JScrollPane(table, v, h);
contentPane.add(jsp, BorderLayout.CENTER);
}
}

```

- a. Error in statement in which JTable is created
- b. Error in statement in which JScrollPane is created
- c. Error in statement in which applet tag is declared
- d. None of the above

**Q16. What will be the output of the following program?**

**2M**

```

import java.awt.*;
import java.applet.*;
public class LayoutDemo5 extends Applet
{
public void init()
{
int i,j,k,n=4;

setLayout(new BorderLayout());
Panel p1=new Panel();
Panel p2=new Panel();

p1.setLayout(new FlowLayout());
p1.add(new TextField(20));
p1.add(new TextField(20));

p2.setLayout(new GridLayout(5,3));
p2.add(new Button("OK"));
p2.add(new Button("Submit"));
add(p1, BorderLayout.EAST);
add(p2, BorderLayout.WEST);
}
}
/*<applet code=LayoutDemo5.class width=300 height=400>
</applet>*/

```

- A. The output is obtained in Frame with two layouts: Frame layout and Flow Layout.
- B. The output is obtained in Applet with two layouts: Frame layout and Flow Layout.
- C. The output is obtained in Applet with two layouts: Frame layout and Border Layout.
- D. The output is obtained in Applet with two layouts: Border layout and Flow Layout.

Scheme – G  
Sample Question Paper Unit Test 2

Course Name: - Computer Engineering Group  
Course Code:- CO/CD/CM/CW/IF  
Semester: - Sixth  
Subject Title: - Advanced Java Programming  
Marks: - 25 Marks

**17625**

---

**Q 1. Which of these methods of DatagramPacket is used to find the port number? 1M**

- A. port()
- B. getPort()
- C. findPort()
- D. recievePort()

**Q2. In the format of URL what is the last part? 1M**

- A. Protocol.
- B. File path.
- C. Port number.
- D. Host name.

**Q3. . . . . . is a full form of SQL. 1M**

- A. Standard query language
- B. Sequential query language
- C. Structured query language
- D. Server side query language

**Q4. Prepared Statement object in JDBC used to execute..... queries. 1M**

- A. Executable
- B. Simple
- C. High level
- D. Parameterized

**Q5. What is servlet? 1M**

- A. Servlets are small program used for developing and executing web applications.
- B. Servlets are small program used for database applications
- C. Servlets are used for intranet programming
- D. Servlets are programs written in C and C++

**Q6. Which of these is a return type of getAddress() method of DatagramPacket class?** **2M**

- A. DatagramPacket
- B. DatagramSocket
- C. InetAddress
- D. ServerSocket

**Q.7 Select the proper method to retrieve the host name of local machine.** **2M**

- A. static InetAddressgetLocalHost( )throws UnknownHostException
- B. static InetAddressgetName(String hostName)throws UnknownHostException
- C. static InetAddress[ ] getAllByName(String hostname throws UnknownHostException
- D. string getHostAddress()

**Q8. Select the proper constructor of URL class.** **2M**

- A. URL(String protocolName, String hostName, intport, String path)
- B. URL(String urlSpecifier)
- C. URL(String protocolName, String hostName, String path)
- D. All of above

**Q9. Rearrange the steps to connect to the database in SQL.** **2M**

- A. Create the connection object
  - B. Execute the query
  - C. Close the connection object
  - D. Register the driver class
  - E. Create the statement object
- A. a-b-c-d-e
  - B. d-a-e-b-c
  - C. e-d-c-b-a
  - D. d-a-c-b-e

**Q10. What is meaning of following statement.**  
**Res.setContentType("text/html");** **2M**

- A. Browser will interpret content as source code.
- B. Browser will take contents as plain text
- C. Browser will interpret content as HTML source code.
- D. Browser will interpret content as Java source code.

**Q11. Consider the following program what will be displayed in the output?** **2M**

```
import java.net.*;
class myAddress
{
    public static void main (String args[])
    {
```

```

try
{
    InetAddress address = InetAddress.getLocalHost();
    System.out.println(address);
}
catch (UnknownHostException e)
{
    System.out.println("Could not find this computer's address.");
}
}
}
}

```

- A. The internet address of the server
- B. The internet address of the client
- C. The internet address of the host
- D. The internet address of any other PC

**Q12. Choose the correct option to establish a connection to database named student and display its contents.**

**2M**

**A.**

```

import java.sql.*;
class Ddemo1
{
    public static void main(String args[]) throws Exception
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection c=DriverManager.getConnection("jdbc:odbc:ODSN", " ", " ");
        Statement s=c.createStatement();
        ResultSet rs=s.executeQuery("select *from StudTable");
        System.out.println("Name" + "\t" + "Roll_No" + "\t" + "Avg");
        while(rs.next())
        {
            System.out.println(rs.getString(1)+"\t"+rs.getInt(2)+"\t\t"+rs.getDouble(3));
        }
        s.close();
        c.close();
    }
}

```

**B.**

```

import java.sql.*;
class Ddemo1
{
    public static void main(String args[]) throws Exception

```

```

    {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection c=DriverManager.getConnection("jdbc:odbc:ODSN", " ", " ");
    ResultSet rs=s.executeQuery("select *from StudTable");
    System.out.println("Name" + "\t " + "Roll_No" + "\t " + "Avg");
    while(rs.next())
    {
    System.out.println(rs.getString(1)+"\t "+rs.getInt(2)+"\t "+rs.getDouble(3));
    }
    s.close();
c.close();
}
}

```

C.

```

import java.sql.*;
class Ddemo1
{
    public static void main(String args[]) throws Exception
    {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Statement s=c.createStatement();
    ResultSet rs=s.executeQuery("select *from StudTable");
    System.out.println("Name" + "\t " + "Roll_No" + "\t " + "Avg");
    while(rs.next())
    {
    System.out.println(rs.getString(1)+"\t "+rs.getInt(2)+"\t "+rs.getDouble(3));
    }
    s.close();
    c.close();
    }
}

```

D.

```

import java.sql.*;
class Ddemo1
{
    public static void main(String args[]) throws Exception
    {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection c=DriverManager.getConnection("jdbc:odbc:ODSN", " ", " ");
    Statement s=c.createStatement();
    ResultSet rs=s.executeQuery("select *from StudTable");

```

```

System.out.println("Name" + "\t" + "Roll_No" + "\t" + "Avg");
while(rs.next())
{
System.out.println(rs.getString(1)+"\t"+rs.getInt(1)+"\t"+rs.getDouble(1));
}
s.close();
c.close();
}
}

```

**Q13.Consider the following program**

**2M**

```

import java.sql.*;
public class db15
{
public static void main(String args[])throws Exception
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection c =DriverManager.getConnection("jdbc:odbc:MyDSN","","");
PreparedStatement s=c.prepareStatement( "update db3 set Name=? where Roll_no=?");
Statement s=c.createStatement( );
s.setString(1,*);
s.setString(2,*);
s.setString(3,*);
s.executeUpdate();

ResultSet rs=s.executeQuery("select* from db3");
System.out.println("Name"+"t"+"Roll no"+"t"+"Avg");
while(rs.next())
{
System.out.println(rs.getString(1)+"t"+rs.getInt(2)+"t"+rs.getDouble(3));
}
s.close();
c.close();
}
}

```

What should be the input in the position of '?'

- A. Command line argument
- B. Array
- C. Vector
- D. Integer

**Q14.Choose the correct syntax for getConnection() method.**

**2M**

- A. public static Connection getConnection(String url, String password) throws SQLException
- B. public static Connection getConnection(String name, String password) throws SQLException

- C. public static Connection getConnection(String url, String name, String password) throws SQLException
- D. public static Connection getConnection(String url, String name) throws SQLException

**Q15. Observe the code and select the proper output.**

**2M**

Contents Of ColorGet.html File :

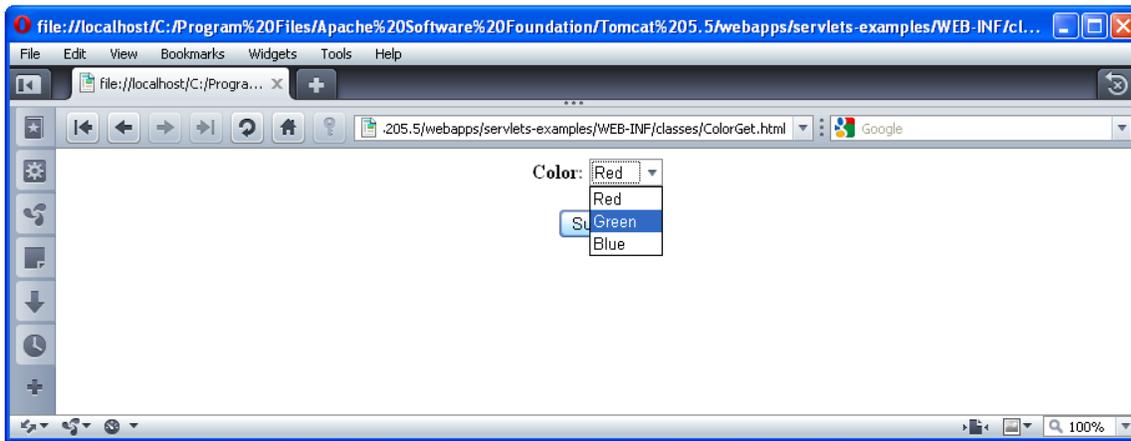
```
<html>
<body>
<center>
<form name="Form1"
action="http://localhost:8080/servlets-examples/ColorGetServlet">
<B>Color:</B>
<select name="color" size="1">
<option value="Red">Red</option>
<option value="Green">Green</option>
<option value="Blue">Blue</option>
</select>
<br><br>
<input type=submit value="Submit">
</form>
</body>
</html>
```

Contents Of ColorGetServlet.java File :

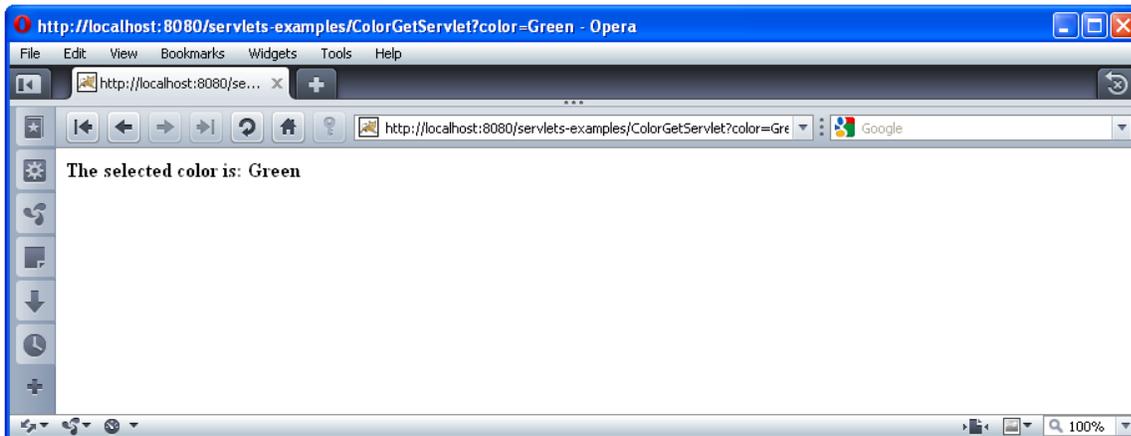
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ColorGetServlet extends HttpServlet
{
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
    {
        String color = request.getParameter("color");
        response.setContentType("text/html");
        PrintWriter pw = response.getWriter();
        pw.println("<B>The selected color is: ");
        pw.println(color);
        pw.close();
    }
}
```

**Output of above program is**  
**Opening : ColorGet.html File**



After clicking on Submit button :



- A. Output shown is correct
- B. Output shown is not correct
- C. Output shown has some errors
- D. Output will be something other than above output

16. A JSP page called test.jsp is passed a parameter name in the URL using `http://localhost/test.jsp?name="John"`. The test.jsp contains the following code.

```
<%! String myName=request.getParameter();%>  
<% String test= "welcome" + myName; %>  
<%= test%>
```

- A. The program prints "Welcome John"
- B. The program gives a syntax error because of the statement `<%! String myName=request.getParameter();%>`
- C. The program gives a syntax error because of the statement `<% String test= "welcome" + myName; %>`
- D. The program gives a syntax error because of the statement `<%= test%>`

# MSBTE Sample Question Paper

Course Name: Computer Engineering Group

Course Code: CO/CM/CD/IF/CW

Subject:- ADVANCED JAVA PROGRAMMING

Marks:- 100

Semester: SIXTH

Subject Code :-(17085)

Hours :- 2 Hrs

---

## Remember Level

1M

### Chapter 1

1. Which class can be used to represent the Checkbox with a textual label that can appear in a menu?

- A. MenuBar
- B. MenuItem
- C. CheckboxMenuItem
- D. Menu

2. Which are various AWT controls from following?

- A. Labels, Push buttons, Check boxes, Choice lists.
- B. Text components, Threads, Strings, Servlets, Vectors
- C. Labels, Strings, JSP, Netbeans, Sockets
- D. Push buttons, Servlets, Notepad, JSP

3. JPanel and Applet use \_\_\_\_\_ as their default layout

- A. FlowLayout
- B. GridLayout
- C. BorderLayout
- D. GridBagLayout

4. Which of the following is true about AWT and Swing Component?

- A. AWT Components create a process where as Swing Component create a thread
- B. AWT Components create a thread where as Swing Component create a process
- C. Both AWT and Swing Component create a process
- D. Both AWT and Swing Component create a thread

5. Which of these methods cannot be called on JLabel object?

- A. setIcon()
- B. getText()
- C. setLabel()
- D. setBorderLayout()

6. \_\_\_\_\_ pane can be used to add component to container

- A. Glass
- B. Content
- C. Container
- D. All of above

**7. Which of the following is not a constructor of JTree**

- A. JTree(Object obj[])
- B. JTree(TreeNodetn)
- C. JTree(Vector v)
- D. JTree(int x)

**8. Swing Components are\_\_\_\_\_**

- A. Platform dependent
- B. Platform Independent
- C. Both a & b
- D. Platform oriented

**Chapter 2**

**1. Which of these methods is used to obtain the object that generated a WindowEvent?**

- A. getMethod()
- B. getWindow()
- C. getWindowEvent()
- D. getWindowObject()

**2. Which of these methods is used to get x coordinate of the mouse?**

- A. getX()
- B. getXCoordinate()
- C. getCoordinateX()
- D. getPointX()

**3. Which of these are constants defined in WindowEvent class?**

- A. WINDOW\_ACTIVATED
- B. WINDOW\_CLOSED
- C. WINDOW\_DEICONIFIED
- D. All of the mentioned

**4. Which of these is super class of WindowEvent class?**

- A. WindowEvent
- B. ComponentEvent
- C. ItemEvent
- D. InputEvent

**Chapter 3**

**1. Which of these is a return type of getAddress method of DatagramPacket class?**

- A. DatagramPacket
- B. DatagramSocket
- C. InetAddress
- D. ServerSocket

**2. In the format for defining the URL what is the last part?**

- A. Protocol.
- B. File path.
- C. Port number.

D. Host name.

**3. What is the first part of URL address?**

A. Host name.

B. Port number.

C. File path.

D. Protocol.

**4. Which of these methods of DatagramPacket is used to obtain the byte array of data contained in a datagram?**

A. getData()

B. getBytes()

C. getArray()

D. receiveBytes()

## Chapter 4

**1. Native – protocol pure Java converts ..... in to the ..... used by DBMSs directly.**

A. JDBC calls, network protocol

B. ODBC class, network protocol

C. ODBC class, user call

D. JDBC calls, user call

**2. The JDBC-ODBC bridge driver resolves..... and makes equivalent .....**

A. JDBC call, ODBC call

B. ODBC call, ODBC call

C. ODBC call, JDBC call

D. JDBC call, JDBC call

**3. For execution of DELETE SQL query in JDBC, ..... method must be used.**

A. executeQuery()

B. executeDeleteQuery()

C. executeUpdate()

D. executeDelete()

**4. Prepared Statement object in JDBC used to execute..... queries.**

A. Executable

B. Simple

C. High level

D. Parameterized

## Chapter 5

**1. Name the class that includes the getSession() method that is used to get the HttpSession object**

A. HttpServletRequest

B. HttpServletResponse

C. SessionContext

D. SessionConfig

2. A user types the URL `http://www.msbte.com/result.php`. Which HTTP request gets generated? Select the one correct answer

- A. GET method
- B. POST method
- C. HEAD method
- D. PUT method

3. Which of these is a protocol for breaking and sending packets to an address across a network?

- A. TCIP/IP
- B. DNS
- C. Socket
- D. Proxy Server

4. in a web application, running in a webserver, who is responsible for creating request and response object,

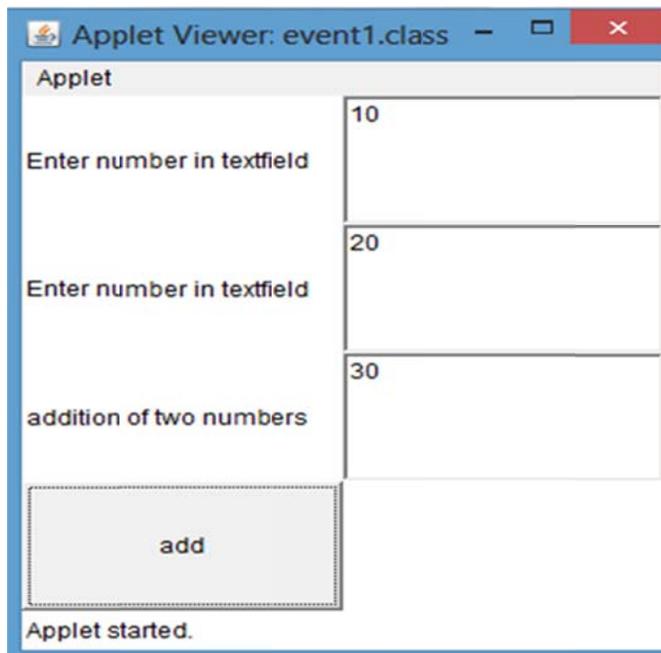
- A. Web server
- B. Servlet
- C. Container
- D. Client

## Understand Level

2M

### Chapter 1

1. Which components are used in the following output?



- A. Label, TextField, Button

- B. Applet, Label
- C. Applet, Button
- D. Grid Layout, Label, Button

**2. What is the purpose of JTable?**

- A. JTable object displays rows of data.
- B. JTable object displays columns of data.
- C. JTable object displays rows and columns of data.
- D. JTable object displays data in Tree form.

**3. Which method is used to display icon on a component?**

- A. rollOverIcon(ImageIcon i)
- B. setIcon(ImageIcon i)
- C. displayIcon(ImageIcon i)
- D. removeIcon (ImageIcon i)

**4. Which components will be needed to get following output?**



- A. Label, TabbedPane, CheckBox
- B. TabbedPane, List, Applet
- C. Panel, TabbedPane, List
- D. Applet, TabbedPane, Panel

**Chapter 2**

**1. Select the missing statement in given code**

```
import java.awt.*;
import java.applet.*;
/*
<applet code="mouse" width=300 height=100>
</applet>
*/
public class mouse extends Applet implements MouseListener, MouseMotionListener
{
String msg = "";
int mouseX = 0, mouseY = 0
public void init()
{
```

```

}
public void mouseClicked(MouseEvent me)
{
mouseX = 0;
mouseY = 10;
msg = "Mouse clicked.";
repaint();
}
public void mouseEntered(MouseEvent me)
{
mouseX = 0;
mouseY = 10;
msg = "Mouse entered.";
repaint();
}
public void mouseExited(MouseEvent me)
{
mouseX = 0;
mouseY = 10;
msg = "Mouse exited.";
repaint();
}
public void mousePressed(MouseEvent me)
{
mouseX = me.getX();
mouseY = me.getY();
msg = "Down";
repaint();
}
public void mouseReleased(MouseEvent me)
{
mouseX = me.getX();
mouseY = me.getY();
msg = "Up";
repaint();
}
public void mouseDragged(MouseEvent me)
{
mouseX = me.getX();
mouseY = me.getY();
msg = "*";
showStatus("Dragging mouse at " + mouseX + ", " + mouseY);
repaint();
}
public void mouseMoved(MouseEvent me)
{
showStatus("Moving mouse at " + me.getX() + ", " + me.getY());
}
public void paint(Graphics g)

```

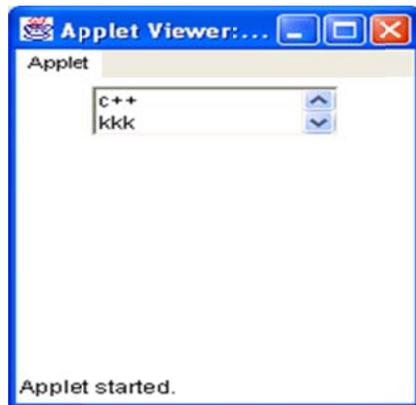
```
{  
g.drawString(msg, mouseX, mouseY);  
}  
}
```

- a) `addMouseListener(this);`
- b) `addMouseListener(this);`
- c) `import java.awt.event.*;`
- d) all of above

**2. Select the proper output for following code**

```
import java.awt.*;  
import java.applet.*;  
public class list2 extends Applet  
{  
public void init()  
{  
List l= new List(2,true);  
l.add("java");  
l.add("c++");  
l.add("kkk");  
add(l);  
}  
}  
/*<applet code=list2.class height=200 width=200>  
</applet>*/
```

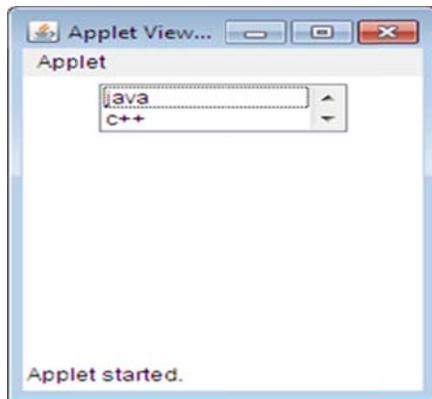
a)



b)



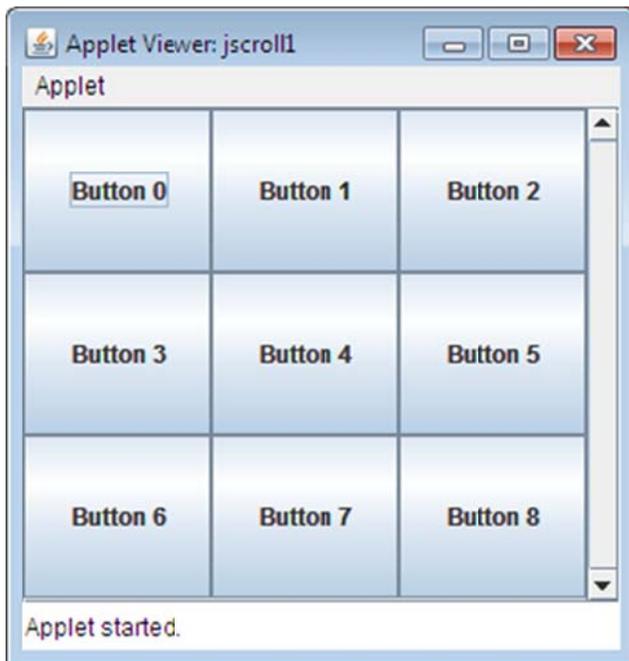
c)



d)



3. To get the following output complete the code given below.



```

import java.awt.*;
import javax.swing.*;
/*
<applet code="jscroll" width=300 height=250>
</applet>
*/
public class jscroll extends JApplet
{
public void init()
{
Container contentPane = getContentPane();
contentPane.setLayout(new BorderLayout());
}
}
int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS;
int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
JScrollPanejsp = new JScrollPane(jp, v, h);
contentPane.add(jsp, BorderLayout.CENTER);
}
}

```

A) Container contentPane = getContentPane();  
contentPane.setLayout(new GridLayout());

B) JPaneljp = new JPanel();  
jp.setLayout(new GridLayout(20, 20));

C) int b = 0;  
for(int i = 0; i < 20; i++) {  
for(int j = 0; j < 20; j++) {

```

        jp.add(new JButton("Button " + b));
        ++b;

D) JPanel jp = new JPanel();
   jp.setLayout(new GridLayout(3,3));
   int b = 0;
   for(int i = 0; i <3; i++)
   {
       for(int j = 0; j <3; j++)
       {
           jp.add(new JButton("Button " + b));
           ++b;
       }
   }

```

## Chapter 3

### 1. Select the proper method to retrieve the host name of local machine

- A. static InetAddress getLocalHost( ) throws UnknownHostException
- B. static InetAddress getByname(String hostName) throws UnknownHostException
- C. static InetAddress[ ] getAllByName(String hostname throws UnknownHostException
- D. string getHostAddress()

### 2. Select the proper constructor of URL class

- A. URL(String protocolName, String hostName, int port, String path)
- B. URL(String urlSpecifier)
- C. URL(String protocolName, String hostName, String path)
- D. All of above

### 3. Select the proper constructor of server socket

- A. ServerSocket(int port, int maxQueue)
- B. Socket(InetAddress address, int port)
- C. Socket(int port)
- D. ServerSocket()

### 4. What will be displayed in the output?

```

import java.net.*;
class myAddress
{
    public static void main (String args[])
    {
        try
        {
            InetAddress address = InetAddress.getLocalHost();
            System.out.println(address);
        }
        catch (UnknownHostException e)

```

```
{
System.out.println("Could not find this computer's address.");
}
}
}
```

- A. The internet address of the server
- B. The internet address of the client
- C. The internet address of the host
- D. The internet address of any other PC

## Chapter 4

**1. executeQuery() method returns \_\_\_\_\_**

- A. Single row
- B. ResultSet object
- C. Single Column
- D. Database Table

**2. PreparedStatement interface extends \_\_\_\_\_ interface**

- A. Connection
- B. Statement
- C. ResultSet
- D. Driver

**3. executeUpdate() method returns \_\_\_\_\_**

- A. Single row
- B. ResultSet object
- C. Integer
- D. Single Column

## Chapter 5

**1. Identify correct syntax of service() method of servlet class**

- A. void service(ServletRequest req, ServletResponse res)
- B. void service(ServletResponse res ServletRequest req, )
- C. void service(ServletRequest req, ServletRequest req )
- D. void service(ServletResponsereq, ServletResponse res)

**2. Advantage of JSP over Servlet is \_\_\_\_\_**

- A. JSP is web page and servlets are Java programs
- B. JSP is web page scripting language and servlets are Java programs
- C. JSP is web page scripting language and servlets are simple programs
- D. JSP is program and servlets are scripting language

**3. Difference between doGet() and doPost() methods is\_\_\_\_\_ . Select any of given options**

- A. In doGet() the parameters are appended to the URL and sent along with header information.
- B. In doPost(),will send the information through a socket back to the webservice and it won't show up in the URL bar.
- C. doGet() is a request for information;
- D. doPost() provides information (such as placing an order) that the server is expected to remember

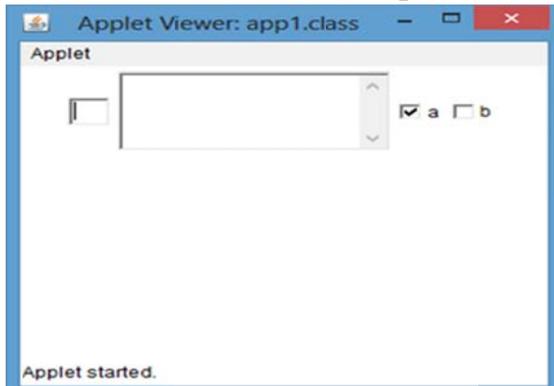
- A. All above are valid differences**
- B. Only A and B**
- C. Only C and D**
- D. A, B, C are valid differences.**

**Apply Level**

**2M**

### **Chapter 1**

**1. Choose the correct sequence for the following output**



- A.**  

```
import java.awt.*;  
import java.applet.*;  
public class app1 extends Applet  
{  
    public void init()  
{
```

```

TextField tf = new TextField();
TextArea t1 = new TextArea(3,20);
Checkbox c = new Checkbox("a",true);
Checkbox c1 = new Checkbox("b",false);
add(t1);
add(c);
add(tf);
add(c1);
}
}
/*<applet code=app1.class width=200 height=200>
</applet>*/

```

B.

```

import java.awt.*;
import java.applet.*;
public class app1 extends Applet
{
public void init()
{
TextField tf = new TextField();
TextArea t1 = new TextArea(3,20);
Checkbox c = new Checkbox("a",true);
Checkbox c1 = new Checkbox("b",false);
add(tf);
add(t1);
add(c);
add(c1);
}
}
/*<applet code=app1.class width=200 height=200>
</applet>*/

```

C.

```

import java.awt.*;
import java.applet.*;
public class app1 extends Applet
{
public void init()
{
TextField tf = new TextField();
TextArea t1 = new TextField();
Checkbox c = new Checkbox("a",true);
Checkbox c1 = new Checkbox("b",false);

```

```

add(tf);
add(t1);
add(c);
add(c1);
}
}

```

D. All of above

**2. Consider the following program. Find which statement contains error.**

```

import java.awt.*;
import javax.swing.*;
/*
<applet code="JTableDemo" width=400 height=200>
</applet>
*/
public class JTableDemo extends JApplet
{
public void init() {
Container contentPane = getContentPane();
contentPane.setLayout(new BorderLayout());
final String[] colHeads = { "emp_Name", "emp_id", "emp_salary" };
final Object[][] data = {
{ "Ramesh", "111", "50000" },
{ "Sagar", "222", "52000" },
{ "Virag", "333", "40000" },
{ "Amit", "444", "62000" },
{ "Anil", "555", "60000" },
};
JTable table = new JTable(data);
int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
JScrollPane jsp = new JScrollPane(table, v, h);
contentPane.add(jsp, BorderLayout.CENTER);
}
}

```

- A. Error in statement in which JTable is created
- B. Error in statement in which JScrollPane is created
- C. Error in statement in which applet tag is declared
- D. None of the above

**3. Select the proper command to run the following code**

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.applet.*;

```

```

/*
<applet code="combodemo11" width=300 height=100>
</applet>
*/
public class combodemo11 extends JApplet
{
public void init()
{
Container co = getContentPane();
co.setLayout(new FlowLayout());
JComboBox jc=new JComboBox();
jc.addItem("cricket");
jc.addItem("football");
jc.addItem("hockey");
jc.addItem("tennis");
co.add(jc);
}
}

```

- A. Javac combodemo11.java
- B. java combodemo11
- C. appletviewer combodemo11.java
- D. All of above**

**3. Observe the following code.**

**What will be the output of the program?**

```

import java.awt.*;
import java.applet.*;
public class LayoutDemo5 extends Applet
{
public void init()
{
inti,j,k,n=4;
setLayout(new BorderLayout());
Panel p1=new Panel();
Panel p2=new Panel();

p1.setLayout(new FlowLayout());
p1.add(new TextField(20));
p1.add(new TextField(20));

p2.setLayout(new GridLayout(5,3));
p2.add(new Button("OK"));
p2.add(new Button("Submit"));

```

```

add(p1, BorderLayout.EAST);
add(p2, BorderLayout.WEST);
}
}
/*<applet code=LayoutDemo5.class width=300 height=400>
</applet>*/

```

- E. The output is obtained in Frame with two layouts: Frame layout and Flow Layout.
- F. The output is obtained in Applet with two layouts: Frame layout and Flow Layout.
- G. The output is obtained in Applet with two layouts: Frame layout and Border Layout.
- H. The output is obtained in Applet with two layouts: Border layout and Flow Layout.

## Chapter 2

### 1. Select proper code for given output



**A.**

```

import java.awt.*;
import java.applet.*;
public class choice11 extends Applet
{
    public void init()
    {
        Choice os=new Choice();
        os.add("wnn18");
        os.add("wnn xp");
        os.add("wnn nt");
        os.add("win 2000");
        add(os);
    }
}
/*<applet code="choice11" height=200 width=300>
</applet>*/

```

```

B.
import java.awt.*;
import java.applet.*;
public class choice11 extends Applet
{
public void init()
{
Choice os=new Choice();
os.add("wnn18");
os.add("wnnxp");
add(os);
}
}
/*<applet code="choice11" height=200 width=300>
</applet>*/

```

```

C.
import java.awt.*;
import java.applet.*;
public class choice11 extends Applet
{
public void init()
{
Choice os=new Choice();
os.add("wnn18");
os.add("wnnxp");
os.add("wnnnt");
os.add("win 2000");
add(os);
}
}

```

```

D.
import java.awt.*;
import java.applet.*;
public class choice11 extends Applet
{
public void init()
{
Choice os=new Choice();
os.add("wnn18");
os.add("wnnxp");
os.add("wnnnt");
os.add("win 2000");
}
}
/*<applet code="choice11" height=200 width=300>
</applet>*/

```

2. Select the missing statement in the program to get the following output

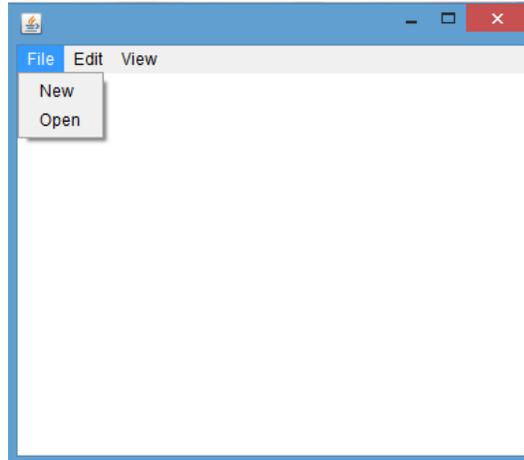


```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
/*
<applet code="combodemo" width=300 height=100>
</applet>
*/
public class combodemo extends JApplet
implements ItemListener
{
JLabel jl;
ImageIcon france, germany, italy, japan;
public void init()
{
Container contentPane = getContentPane();
contentPane.setLayout(new FlowLayout());
JComboBox jc = new JComboBox();
jc.addItem("France");
jc.addItem("Germany");
jc.addItem("Italy");
jc.addItem("Japan");
jc.addItemListener(this);
contentPane.add(jc);
contentPane.add(jl);
}
public void itemStateChanged(ItemEvent ie)
{
String s = (String)ie.getItem();
jl.setIcon(new ImageIcon(s + ".gif"));
}
```

```
}  
}
```

- A. `jl = new JLabel(new ImageIcon("star.gif"));`
- B. `jl = new JLabel("star.gif");`
- C. `jl = new JLabel( ImageIcon("star.gif"));`
- D. `JLabel(new ImageIcon("star.gif"));`

**3. Select the missing statement in the program for following output**

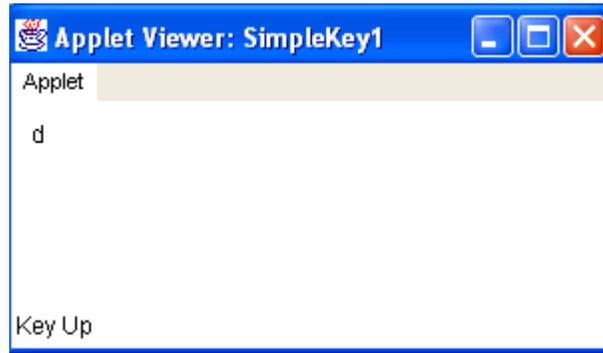


```
import java.awt.*;  
public class MenuDemo extends Frame  
{  
    public static void main(String args[])  
    {  
        MenuDemo m = new MenuDemo();  
        m.setVisible(true);  
        MenuBar mbr = new MenuBar();  
        m.setMenuBar(mbr);  
        Menu filemenu = new Menu("File");  
        Menu editmenu = new Menu("Edit");  
        Menu viewmenu = new Menu("View");  
        mbr.add(filemenu);  
        mbr.add(editmenu);  
        MenuItem new1 = new MenuItem("New");  
        MenuItem open1 = new MenuItem("Open");  
        filemenu.add(new1);  
        filemenu.add(open1);  
    }  
}
```

- A. `mbr.add(view);`
- B. `mbr.add(menu);`

- C. mbr.add(vieweditmenu);
- D. mbr.add(viewmenu);

4. Consider the following output. Find the missing statement in the program.



```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;
/*
<applet code="SimpleKey1" width=300 height=100>
</applet>
*/
public class SimpleKey1 extends JApplet
implements KeyListener
{
String msg = "";
int X = 10, Y = 20; public void init()
{
addKeyListener(this);
requestFocus();
}
public void keyPressed(KeyEvent ke)
{
showStatus("Key Down");
}
public void keyReleased(KeyEvent ke)
{
showStatus("Key Up");
}
public void keyTyped(KeyEvent ke)
{
msg += ke.getKeyChar();
repaint();
}
public void paint(Graphics g)
{
g.drawString(msg, X, Y);
}
}
```

- }  
**A.** Missing Semicolon  
**B.** }  
**C.** {  
**D.** ()

**5. For the following code select the method that can be used to handle event.**

```
import java.awt.event.*;
import java.awt.*;
import java.applet.*;
public class checkbackg extends Applet implements ItemListener
{
    Checkbox m1,m2,m3;
    public void init()
    {
        m1=new Checkbox("A");
        m2=new Checkbox("B");
        m3=new Checkbox("C");
        add(m1);
        add(m2);
        add(m3);
        m1.addItemListener(this);
        m2.addItemListener(this);
    }

    public void _____(ItemEvent ie)
    {
        if(ie.getSource()==m1)
            setBackground(Color.red);
        if(ie.getSource()==m2)
            setBackground(Color.green);
    }
}
/*<applet code=checkbackg.class height=150 width=150>
</applet>*/
```

- A. actionPerformed(ActionEvent ae)  
 B. itemStateChanged(ActionEvent ie)  
 C. itemStateChanged(ItemEvent ie)  
 D. adjustmentPerformed(AdjustmentEvent ae)

### Chapter 3

**1. Consider the following program**

**What will be displayed in the output?**

```
import java.net.*;
```

```

class myAddress
{
public static void main (String args[])
{
try
{
    InetAddress address = InetAddress.getLocalHost();
System.out.println(address);
}
catch (UnknownHostException e)
{
System.out.println("Could not find this computer's address.");
}
}
}
}

```

- A. The internet address of the server
- B. The internet address of the client
- C. The internet address of the host
- D. The internet address of any other PC

**2. Consider the following program**

**What correction should be done in the program to get correct output?**

```

import java.net.*;
import java.io.*;

public class URLTest
{
public static void main(String args[]) throws MalformedURLException
{
URL url = new URL("http://www.msbt.com/download");
System.out.println("Protocol:"+ url1.getProtocol());
System.out.println("Port:"+ url1.getPort());
System.out.println("Host:"+ url1.getHost());
System.out.println("File:"+ url1.getFile());
}
}
}

```

- A. Exception type is wrong.
- B. Class should not be public.
- C. Creation of object is not correct.
- D. Use of created object not correct

**Chapter 4**

**1. Consider the following program.**

**What should be the correction done in the program to get correct output?**

```

import java.sql.*;
class Ddemo1

```

```

{
public static void main(String args[]) throws Exception
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection c=DriverManager.getConnection("jdbc:odbc:ODSN", " ", " ");
Statement s=c.createStatement();
ResultSet rs=s.executeQuery("select *from StudTable");
System.out.println("Name" + "\t" + "Roll_No" + "\t" + "Avg");
while(rs.next())
{
System.out.println(rs.getString(1)+"\t"+rs.getInt(2)+"\t\t"+rs.getDouble(3));
s.close();
c.close();
}
}
}

```

- A. Missing semicolon
- B. Missing {
- C. Missing }
- D. Missing statement.

**2. Consider the following program.**

**What should be the correction done in the program to get correct output?**

```

import java.sql.*;
class Ddemo1
{
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection c=DriverManager.getConnection("jdbc:odbc:ODSN", " ", " ");
Statement s=c.createStatement();
ResultSet rs=s.executeQuery("select *from StudTable");
System.out.println("Name" + "\t" + "Roll_No" + "\t" + "Avg");
while(rs.next())
{
System.out.println(rs.getString(1)+"\t"+rs.getInt(2)+"\t\t"+rs.getDouble(3));
}
s.close();
c.close();
}
}
}

```

- A. Missing semicolon
- B. Missing {
- C. Missing }
- D. Missing statement.

**3. Consider the following program.**

**What should be the correction done in the program to get correct output?**

```

import java.sql.*;
class Ddemo1
{
public static void main(String args[]) throws Exception;
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection c=DriverManager.getConnection("jdbc:odbc:ODSN"," "," ");
Statement s=c.createStatement();
ResultSet rs=s.executeQuery("select *from StudTable");
System.out.println("Name" + " \t " + "Roll_No" + " \t " + "Avg");
while(rs.next())
{
System.out.println(rs.getString(1)+" \t "+rs.getInt(2)+" \t \t"+rs.getDouble(3));
}
s.close();
c.close();
}
}

```

- A. Error in main()
- B. Error in loop
- C. Error in connection statement
- D. Error in close()

**4. Consider the following program.**

**What should be the correction done in the program to get correct output?**

```

class Ddemo1
{
public static void main(String args[]) throws Exception
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection c=DriverManager.getConnection("jdbc:odbc:ODSN"," "," ");
Statement s=c.createStatement();
ResultSet rs=s.executeQuery("select *from StudTable");
System.out.println("Name" + " \t " + "Roll_No" + " \t " + "Avg");
while(rs.next())
{
System.out.println(rs.getString(1)+" \t "+rs.getInt(2)+" \t \t"+rs.getDouble(3));
}
s.close();
c.close();
}
}

```

- A. Missing semicolon
- B. Missing {
- C. Missing }
- D. Missing package statement.

## 5. Consider the following program

Select the statement that should be added to the program to get correct output.

```
import java.sql.*;
public class db15
{
public static void main(String args[])throws Exception
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection c =DriverManager.getConnection("jdbc:odbc:MyDSN","","");
PreparedStatement s=c.prepareStatement( "update db3 set Name=? where Roll_no=?");
Statement s=c.createStatement( );
s.setString(1,args[0]);
s.setString(2,args[1]);
s.setString(3,args[2]);
ResultSet rs=s.executeQuery("select* from db3");
System.out.println("Name"+"\\t"+"Roll no"+"\\t"+"Avg");
while(rs.next())
{
System.out.println(rs.getString(1)+"\\t"+rs.getInt(2)+"\\t"+rs.getDouble(3));
}
s.close();
c.close();
}
}
```

- A. s.executeUpdate()
- B. c.createStatement( )
- C. s.close()
- D. c.close()

## Chapter 5

1. Choose missing statements in following code from given options.

```
public class session1 extends HttpServlet
{
public void doGet(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException
{
HttpSession hs = request.getSession(true);
response.setContentType("text/html");
PrintWriter pw = response.getWriter();
pw.print("<B>");
Date date = (Date)hs.getAttribute("date");
if(date != null) {
pw.print("Last access: " + date + "<br>");
}
date = new Date();
hs.setAttribute("date", date);
}
```

```

pw.println("Current date: " + date);
}
}

```

- A. import java.io.\*; import java.util.\*; import javax.servlet.\*; import javax.servlet.http.\*;
- B. import java.Vector.\*; import java.Thread.\*; import javax.servlet.\*;
- C. import javax.servlet.http.\*; import java.String.\*; import java.Vector;
- D. import javax.servlet.http.\*; import java.Thread.\*; import javax.Client.\*;

**2. In following Java program fill statement showing \*\*\*.Select any one option from given options**

```

import javax.servlet.*;
import javax.servlet.http.*;
public class AddCookieServlet extends HttpServlet
{
public void doPost(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException
{
String data = request.getParameter("data");
Cookie cookie = *****
response.addCookie(cookie);
response.setContentType("text/html");
PrintWriter pw = response.getWriter();
pw.println("<B>MyCookie has been set to");
pw.println(data);
pw.close();
}
}

```

- A. new Cookie("MyCookie", data);
- B. new Cookie("MyCookie", data1);
- C. new Cookie("MyCookie", data2);
- D. new Cookie("MyCookie", database);

**3. Consider the following program. Identify the exception that might be thrown**

```

import java.net.*;
class URLLDemo
{
public static void main(String args[]) throws _____
{
URL netAddress= new URL("http://www.sun.com:8080//index.html");
System.out.println("Protocol :"+netAddress.getProtocol());
System.out.println("Port :"+netAddress.getPort());
System.out.println("Host :"+netAddress.getHost());
}
}

```

```
System.out.println("File :"+netAddress.getFile());
}
}
```

- A. IOException
- B. MalformedURLException
- C. Arithmetic Exception
- D. UnknownHostException

**4. Consider the following program. Identify the missing statement from the output.**

```
import java.net.*;
class URLEDemo
{
public static void main(String args[]) throws MalformedURLException
{
URL netAddress= new URL("http://www.sun.com: //index.html");
System.out.println("Protocol :"+netAddress.getProtocol());
System.out.println("Port :"+netAddress.getPort());
System.out.println("Host :"+netAddress.getHost());
System.out.println("File :"+netAddress.getFile());
}
}
```

**Output of the Program**

```
Protocol :http
      Host :www.sun.com
      File :/index.html
```

- A. Port :8080
- B. Port :1024
- C. Port: -1
- D. None of the above

**5. Consider the following program and identify the missing statement.**

```
class URLEDemo
{
public static void main(String args[]) throws MalformedURLException
{
URL netAddress= new URL("http://www.sun.com:/index.html");
System.out.println("Protocol :"+netAddress.getProtocol());
System.out.println("Port :"+netAddress.getPort());
System.out.println("Host :"+netAddress.getHost());
System.out.println("File :"+netAddress.getFile());
}
}
```

- A. Missing semicolon
- B. Missing package statement

- C. Missing initialization
- D. None of the above

### 6.2.3 Answer Key

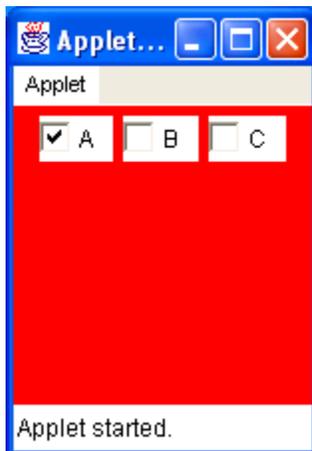
#### Answer key for Unit test I

1. Answer : C: remove()
2. Answer: A: Controls or components allow users to interact with application
3. Answer: A: Windows, Panel, ScrollPane
4. Answer :A: CheckboxGroup
5. Answer: A : An event is an object that describes a state change in a source.
6. Answer: C: addMouseMotionListner()
7. Answer: A getID()
8. Answer: B: List, Button
9. Answer : C: Panel, TabbedPane, List
10. Answer: B:     addMouseListener(this);

```
addMouseMotionListener(this);
import java.awt.event.*;
```

11. Answer:C:AdjustmentEvent

12. Answer: B

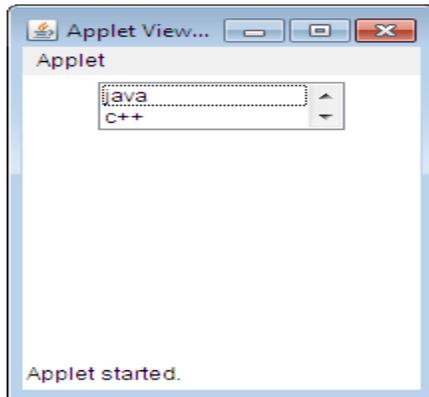


13. Answer: A:

```
import java.awt.*;
import java.applet.*;
public class choice11 extends Applet
{
public void init()
{
Choice os=new Choice();
os.add("wnn18");
os.add("wnnxp");
os.add("wnmnt");
os.add("win 2000");
add(os);
}
}
```

```
/*<applet code="choice11" height=200 width=300>  
</applet>*/
```

14. Answer: C:



15. Answer: A: **Error in statement in which JTable is created**

16. Answer: D: The output is obtained in Applet with two layouts: Border layout and Flow Layout.

## Answer key for Unit test II

1. Answer : B: getPort()
2. Answer: B: File path
3. Answer: C: Structured Query Language
4. Answer :D: Parametrized
5. Answer: A : Servlets are small program used for developing and executing web applications.
6. Answer: C: InetAddress
7. Answer: A static InetAddressgetLocalHost( )throws UnknownHostException
8. Answer: D: All of above
9. Answer : B: d-a-e-b-c
10. Answer: C: Browser will interpret content as HTML source code.
11. Answer: C: The internet address of the host
12. Answer: A

```
import java.sql.*;
class Ddemo1
{
    public static void main(String args[]) throws Exception
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection c=DriverManager.getConnection("jdbc:odbc:ODSN"," "," ");
        Statement s=c.createStatement();
        ResultSet rs=s.executeQuery("select *from StudTable");
        System.out.println("Name" + "\t " + "Roll_No" + "\t " + "Avg");
        while(rs.next())
        {
            System.out.println(rs.getString(1)+"\t "+rs.getInt(2)+"\t "+rs.getDouble(3));
        }
        s.close();
        c.close();
    }
}
```

13. Answer: A: Command line argument
14. Answer: C: public static Connection getConnection(String url, String name, String password) throws SQLException
15. Answer: A: Output shown is correct
16. Answer:B:The program gives a syntax error because of the statement  
<%! String myName=request.getParameter();%>

## Answer Key for AJP 100 marks Paper

### Remember Level

#### Chapter 1

1. **Answer: C:** CheckboxMenuItem
2. **Answer: A:** Labels, Push buttons, Check boxes, Choice lists.
3. **Answer: A:** FlowLayout
4. **Answer: A:** AWT Components create a process where as Swing Component create a thread
5. **Answer: D:** setBorderLayout()
6. **Answer: B:** Cotent
7. **Answer: D:** JTree(int x)
8. **Answer: B:** Platform Independent

#### Chapter 2

1. **Answer: B:** getWindow()
2. **Answer: B:** getXCoordinate()
3. **Answer: D:** All of the mentioned
4. **Answer: B:** ComponentEvent

#### Chapter 3

1. **Answer: C:** InetAddress
2. **Answer: B:** File path.
3. **Answer: D:** Protocol.
4. **Answer: A:** getData()

#### Chapter 4

1. **Answer: A:** JDBC calls, network protocol
2. **Answer: A:** JDBC call, ODBC call
3. **Answer: C:** executeUpdate()
4. **Answer: D:** Parameterized

#### Chapter 5

1. **Answer: A:** HttpServletRequest
2. **Answer: A:** GET method
3. **Answer: A:** TCIP/IP
4. **Answer: B:** Servlet

### Understand Level

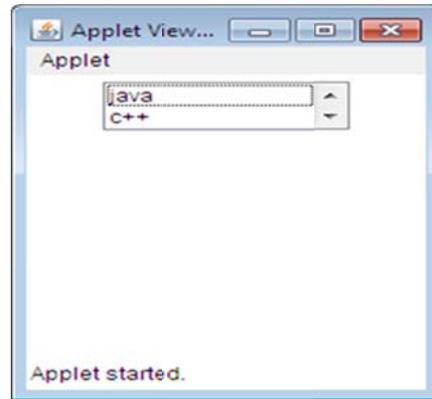
#### Chapter 1

1. **Answer: A:** Label, TextField, Button
2. **Answer: C:** JTable object displays rows and columns of data.
3. **Answer: B:** setIcon(ImageIcon i)
4. **Answer: C:** Panel, TabbedPane, List

#### Chapter 2

1. **Answer: D:** All of the above.

2. Answer: C:



3. Answer: D:

```
JPaneljp = new JPanel();
    jp.setLayout(new GridLayout(3,3));
    int b = 0;
    for(int i = 0; i <3; i++)
    {
        for(int j = 0; j <3; j++)
        {
            jp.add(new JButton("Button " + b));
            ++b;
        }
    }
```

### Chapter 3

1. Answer: A: `static InetAddressgetLocalHost( )` throws `UnknownHostException`.
2. Answer: D: All of above
3. Answer: A: `ServerSocket(int port, int maxQueue)`
4. Answer: C: The internet address of the host

### Chapter 4

1. Answer: B: `ResultSet` object
2. Answer: B: `Statement`
3. Answer: C: `Integer`

### Chapter 5

1. Answer: A: `void service(ServletRequestreq, ServletResponse res)`
2. Answer: B: JSP is web page scripting language and servlets are Java programs
3. Answer: A: All above are valid differences

### Apply Level

### Chapter 1

**1. Answer : B**

```
import java.awt.*;
import java.applet.*;
public class app1 extends Applet
{
    public void init()
    {
        TextField tf = new TextField();
        TextArea t1 = new TextArea(3, 20);
        Checkbox c = new Checkbox("a", true);
        Checkbox c1 = new Checkbox("b", false);
        add(tf);
        add(t1);
        add(c);
        add(c1);
    }
}
/*<applet code=app1.class width=200 height=200>
</applet>*/
```

2. Answer : A: Error in statement in which JTable is created
3. Answer: C: appletviewer combodemo11.java
4. Answer: D: The output is obtained in Applet with two layouts: Border layout and Flow Layout.

**Chapter 2**

**1. Answer: A**

```
import java.awt.*;
import java.applet.*;
public class choice11 extends Applet
{
    public void init()
    {
        Choice os = new Choice();
        os.add("wnn18");
        os.add("wnnxp");
        os.add("wnnnt");
        os.add("win 2000");
        add(os);
    }
}
/*<applet code="choice11" height=200 width=300>
</applet>*/
```

2. Answer: A: `jl = new JLabel(new ImageIcon("star.gif"));`
3. Answer: D: `mbr.add(viewmenu);`
4. Answer C: {
5. Answer: C: `itemStateChanged(ItemEvent ie)`

### Chapter 3

1. Answer: C: The internet address of the host
2. Answer: D: Use of created object not correct

### Chapter 4

1. Answer: C: Missing }
2. Answer: D: Missing statement.
3. Answer: A: Error in main()
4. Answer: D: Missing package statement.
5. Answer: A: `s.executeUpdate()`

### Chapter 5

1. Answer: A: `import java.io.*; import java.util.*; import javax.servlet.*; import javax.servlet.http.*;`
2. Answer: A: `new Cookie("MyCookie", data);`
3. Answer: B: `MalformedURLException`
4. Answer: C: Port: -1
5. Answer: B: Missing package statement