

Question Bank for M.S.B.T.E Exam for Winter 2015

Chapter 2

Classes ,Objects& Methods

Important Questions

Summer-15

1) Define wrapper class. Give the following wrapper class methods with syntax and use(8 Marks)

- To convert integer number to string.
- To convert numeric string to integer number.
- To convert object numbers to primitive numbers using typevalue()

Ans.

Wrapper Class:

Objects like vector cannot handle primitive data types like int, float, long char and double. Wrapper classes are used to convert primitive data types into object types.This object can be stored in vector. Wrapper classes are contained in the java.lang package.Wrapper class has various method.Below are two methods explained.

The some of the wrapper classes are:

Primitive Data type	Wrapper class
boolean	Boolean
int	Integer
char	Character
float	Float

1) To convert integer number to string : Method is toString()

String str = Integer.toString(i) –converts the integer i to string value.

class IntegerExample

```
{
public static void main(String a[]) {
String str = Integer.toString(5);
System.out.println("String value of i "+str );
}
}
```

2) To convert numeric string to integer number: Method is parseInt()

int i = Integer.parseInt(str) – converts the string value of numeric string str to int i.

class StringConversion {

```
public static void main(String a[]) {
String str = "5";
int i = Integer.parseInt(str);
System.out.println("Integer value of str is "+i);
}
}
```

3) To convert object number to primitive number using typevalue() method:

The method here is typeValue(), meaning that type can be the data type of the number. For example : if x is an Integer object with value 10, then it can be stored as primitive int y with the method intValue() as

```
Integer x = new Integer(10);
```

```
int y = x.intValue();
```

Similarly, it can be done with other numeric types as floatValue(),doubleValue() etc.

2) Differentiate vector and array with any 4 points. (4Marks)

Ans.

<u>Array</u>	<u>Vector</u>
Array can accommodate fixed number of elements	Vectors can accommodate unknown number of elements
Arrays can hold primitive data type & objects	Vectors can hold only objects.
All elements of array should be of the same data type. i.e. it can contain only homogeneous elements.	The objects in a vector need not have to be homogeneous.
Syntax : Datatype[] arraname= new datatype[size];	Syntax: Vector objectname= new Vector();
For accessing elements of an array no special methods are available as it is not a class , but derived type.	Vector class provides different methods for accessing and managing Vector elements.

3) State the use of final keyword w. r. t.a method and the variable with suitable example.(4 Marks)

Ans

All variable and methods can be overridden by default in subclass. In order to prevent this, the final modifier is used. Final modifier can be used with variable, method or class.

final variable: the value of a final variable cannot be changed. final variable behaves like class variables and they do not take any space on individual objects of the class.

Eg of declaring final variable: final int size = 100;

final method: making a method final ensures that the functionality defined in this method will never be altered in subclass, ie a final method cannot be overridden.

Eg:

```
public class A
{
final void mymethod(){...};
}
```

```
public class B extends A
{
```

```
final void mymethod(){...}; //forbidden will give error.
}
```

4) Compare string class and StringBuffer class with any four points. (4 Marks)

Ans

Sr. No.	<u>String</u>	<u>StringBuffer</u>
1	String is a major class	StringBuffer is a peer class of String
2	Length is fixed	Length is flexible
3	Contents of object cannot be modified	Contents of can be modified
4	Object can be created by assigning String constants enclosed in double quotes.	Objects can be created by calling constructor of StringBuffer class using <code>_new</code> ,
5	Ex:- <code>String s="abc";</code> <code>String s=new String("abc")</code>	Ex:- <code>StringBuffer s=new StringBuffer("abc");</code>

5) Explain following methods of vector class(6 Marks)

- i. `elementAt ()`
- ii. `addElement ()`
- iii. `removeElement ()`

Ans. Refer Ans 14

Winter-14

6) What is difference between arrays and vectors ? Explain any 2 methods of vector with example.(6Marks)

Ans. Refer Ans 2

Vector class methods :

1) **addElement ()**: the object specified by element is added to the vector.

Syntax:

```
void addElement(Object element)
```

Example:

```
Vector v = new Vector();
v.addElement(new Integer(1)); //add integer object 1 to vector
```

2) **removeElement ()**: Boolean `removeElement(Object element)` : Removes element from the vector. If more than one instance of the specifies object exists in the vector, then it is the first one that is removed. Returns true if successful and false if the object is not found.

Syntax:

```
Boolean removeElement(Object element)
```

Example:

```
Vector v = new Vector();
v.removeElement(new Float(5.45)); //remove object from vector
```

3) **int capacity()**

Returns the current capacity of this vector.

4) **void clear()**

Removes all of the elements from this Vector.

5) **boolean contains(Object e)**

Tests if the specified object is a component in this vector.

6) **void copyInto(Object[] anArray)**

Copies the components of this vector into the specified array.

7) **Object elementAt(int index)**

Returns the component at the specified index.

6) **void insertElementAt(Object obj, int index)**

Inserts the specified object as a component in this vector at the specified index.

7) **void removeElementAt(int index)**

Removes Element At index specified.

8) **int size()**

Returns the number of components in this vector.

7) What is constructor? Demonstrate the use of parameterised constructor with suitable example. (6 Marks)

Ans

- A constructor is a special method which initializes an object immediately upon creation.
- It has the same name as class name in which it resides and it is syntactically similar to any method.
- When a constructor is not defined, java executes a default constructor which initializes all numeric members to zero and other types to null or spaces.
- Once defined, constructor is automatically called immediately when the object is created before new operator completes.
- Constructors do not have return value, not even 'void' because they return the instance of class.
- Constructor called by new operator.

Eg :

class Rect

```
{
int length, breadth;
Rect() //constructor
{
length=4;
breadth=5;
}
public static void main(String args[])
{
Rect r = new Rect();
System.out.println("Area : " +(r.length*r.breadth));
}
}
```

Output :

Area : 20

Parameterized constructor :

Constructor which have arguments are known as parameterized constructor. When constructor method is defined with parameters inside it, different value sets can be provided to different constructor with the same name.

Example

```
class Rect
{
int length, breadth;
Rect(int l, int b) // parameterized constructor
{
length=l;
breadth=b;
}
public static void main(String args[])
{
Rect r = new Rect(4,5); // constructor with parameters
Rect r1 = new Rect(6,7);
System.out.println("Area : " +(r.length*r.breadth));
System.out.println("Area : " +(r1.length*r1.breadth));
}
}
```

Output :

Area : 20

Area : 42

8) Explain abstract class with suitable example.(8 Marks)

Ans

- A class that is declared with abstract keyword is known as abstract class in java. It can have abstract and non-abstract methods (method with body).
- Abstraction is a process of hiding the implementation details and showing only functionality to the user. Another way, it shows only important things to the user and hides the internal details.
- A method must be always redefined in a subclass of an abstract class, as abstract class does not contain method body. Thus abstract makes overriding compulsory.
- Class containing abstract method must be declared as abstract.
- You can not declare abstract constructor, and so, objects of abstract class cannot be instantiated.
- Abstract class has atleast one method declared as abstract.
- Common methods can be put in one abstract class without writing the implementation.

- **Syntax :**

```
abstract class < classname >
```

```
{
```

```
.
```

```
.
```

```
abstract method1(...);
```

```
method2(...);
```

```
.
```

```

.}
Example :
abstract class A
{
abstract void disp();
void show()
{
System.out.println("show method is not abstract");
}
}
class B extends A
{
void disp()
{
System.out.println("inside class B");
}
}
class test
{
public static void main(String args[])
{
B b = new B();
b.disp();
b.show();
}
}

```

Output : show method is not abstract inside class B

9) Explain use of following methods(4 Marks)

- a) indexOf () b) charAt ()
b) substring () d) replace ()

Ans

1. indexOf():

int indexOf(char ch):Returns the index of the first occurrence of the specified character in the string.

Eg.,String s1="sanjay"; s1.index('a');

Output : 1

int indexOf(int ch, int fromIndex): Returns the index of the first occurrence of the specified character, starting the search at the specified fromindex.

2. charAt():

char charAt(int index): Returns the char value at the specified index.

Eg., String s1="vijay"; s1.charAt(2); o/p:j

3. substring():

String substring(int n): Returns a substring of the string from nth character.

Eg.,String s1="vijay"; s1.subString(2);o/p: jay

String substring(int n, int m): gives substring from nth character up to mth not including mth character.

Eg.,String s1="vijay"; s1.substring(0,3);o/p: vij

4. replace():

String replace(char oldchar, char newchar): replaces all appearance of oldchar with newchar.

E.g.,String s1="vijay" ; String s2=s1.replace('a','e');o/p:vijey

10) What is final variable and method ? How it is different from abstract method ?(4 Marks)

Ans

- All variable and methods can be overridden by default in subclass. In order to prevent this, the final modifier is used.
- Final modifier can be used with variable, method or class.

final variable: the value of a final variable cannot be changed. final variable behaves like class variables and they do not take any space on individual objects of the class.

Eg of declaring final variable: final int size = 100;

final method: making a method final ensures that the functionality defined in this method will never be altered in any way, ie a final method cannot be overridden.

Eg:

```
public class A
{
final void mymethod(){...};
}
```

```
public class B extends A
{
final void mymethod(){...}; //forbidden will give error.
}
```

Abstract class

- abstract is a modifier used with methods and classes.
- An abstract method is a method that is declared without an implementation.
- The implementation is to be done in the subclass making overriding compulsory.
- It is opposite of final modifier.
- If a class includes abstract methods, then the class itself must be declared abstract.
- When an abstract class is subclassed, the subclass usually provides implementations for all of the abstract methods in its parent class. However, if it does not, then the subclass must also be declared abstract.

Eg:

```
abstract class Shape
{
int variable1, variable2;
void method1
{
```

```
//implementation
}
abstract void method2();
}
```

11) What is garbage collection and finalize method in Java.(6 Marks)

Ans

Garbage collection - The objects are dynamically allocated in java by the use of the new operator. The objects which are no longer in use are to be destroyed and the memory should be released for later reallocation. Java, unlike C++, handles deallocation of memory automatically. This technique is called garbage collection. When no references to an object exist, that object is assumed to be no longer needed, and the memory occupied by the object can be reclaimed. Garbage collection occurs sporadically during the execution of programs. It will not occur just because one or more objects exist that are no longer used. Different run-time implementations will take varying approaches to garbage collection.

finalize() method: -sometimes an object will need to perform some action when it is destroyed. Eg. If an object holding some non java resources such as file handle or window character font, then before the object is garbage collected these resources should be freed. To handle such situations java provide a mechanism called finalization. In finalization, specific actions that are to be done when an object is garbage collected can be defined. To add finalizer to a class define the finalize() method. The java run-time calls this method whenever it is about to recycle an object. Inside the finalize() method, the actions that are to be performed before an object is to be destroyed, can be defined. Before an object is freed, the java run-time calls the finalize() method on the object. The general form of the finalize() method is:

```
protected void finalize()
{
//finalization code
}
```

Summer-14

12) What is Constructor? Explain overloading with example.(4 Marks)

Ans.

Constructor:

Constructor is a special type of method that is used to initialize the object. Constructor is invoked at the time of object creation. It provides data for the object that is why it is known as constructor. It has same name as class name.

Constructor overloading:

Overloading is having more than one form. Overloading refers to the use of the same thing for different purpose. (Constructor overloading means to define a various constructor but with different signatures)

Constructor overloading is a technique in Java in which a class can have any number of constructors that differ in parameter lists. The compiler differentiates these constructors by taking into account the number of parameters in the list and their type.

Example:

```
Class Sample
{
int x;
Sample()
{
x=0;
}
Sample(int a)
{
x=a;
}
Sample(Sample s)
{
x=s.x;
}
}
```

13) Explain method overloading with example. (4 Marks)**Ans.**

Method Overloading means to define different methods with the same name but different parameters lists and different definitions. It is used when objects are required to perform similar task but using different input parameters that may vary either in number or type of arguments. Overloaded methods may have different return types. It is a way of achieving compile time polymorphism in java.

When we call a method in an object, Java matches up the method name first & then the number & type of parameter to decide which one of the defn to execute. Overloaded methods may have same or different return type.

```
int add( int a, int b) // prototype 1
int add( int a , int b , int c) // prototype 2
double add( double a, double b) // prototype 3
```

Example :

```
class Sample
{
int addition(int i, int j)
{
return i + j ;
}
String addition(String s1, String s2)
{
return s1 + s2;
}
double addition(double d1, double d2)
{
return d1 + d2;
```

```

}
}
class AddOperation
{
public static void main(String args[])
{
Sample sObj = new Sample();
System.out.println(sObj.addition(1,2));
System.out.println(sObj.addition("Hello ","World"));
System.out.println(sObj.addition(1.5,2));
}
}
}

```

14) Explain following method of vector class. (4 Marks)

a)elementAt() b) removeElement c) insertElementAt() d) addElement()

Ans

Refer ans 16

3) insertElementAt() :Adds element to the vector at the location specified by the index.

Syntax : void insertElementAt(Object element, int index)

Example: Vector v = new Vector();

v.insertElementAt(new Integer(5) , 2); //insert integer object in vector at 2nd position.

Winter 13

15) Describe the following methods related to vector (6 Marks)

a)addElement() b)removeElement() c)elementAt()

Ans

a) **addElement ()**: the object specified by element is added to the vector.

Syntax:

void addElement(Object element)

Example:

Vector v = new Vector();

v.addElement(new Integer(1)); //add integer object 1 to vector

b) removeElement(): Boolean removeElement(Object element) : Removes element from the vector.

If more than one instance of the specifies object exists in the vector,then it is the first one that is removed. Returns true if successful and false if the object is not found.

Syntax:

Boolean removeElement(Object element)

Example:

Vector v = new Vector();

v.removeElement(new Float(5.45)); //remove object from vector

c) elementAt(): Returns the element at the location specified by index.

Syntax:

Object elementAt(int index)

Example:

```
Vector v = new Vector();  
v.elementAt(2); //return 2nd element from vector
```

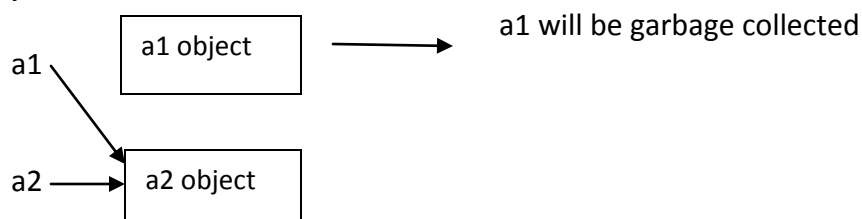
16) What is garbage collection in java? Explain. (4 Marks)

Ans

Garbage Collection. Since objects are dynamically allocated by using the new operator, you might be wondering how such objects are destroyed and their memory released for later reallocation. In some languages, such as C++, dynamically allocated objects must be manually released by use of a delete operator. Java takes a different approach; it handles deallocation for you automatically. The technique that accomplishes this is called garbage collection. It works like this: when no references to an object exist, that object is assumed to be no longer needed, and the memory occupied by the object can be reclaimed. There is no explicit need to destroy objects as in C++. Garbage collection only occurs sporadically (if at all) during the execution of your program. It will not occur simply because one or more objects exist that are no longer used. Furthermore, different Java run-time implementations will take varying approaches to garbage collection, but for the most part, you should not have to think about it while writing your programs.

Example:

```
public class A  
{  
  int p, q;  
  A ()  
  {  
    p = 10;  
    p = 20;  
  }  
}  
class Test  
{  
  public static void main(String args[])  
  {  
    A a1= new A();  
    A a2= new A();  
    a1=a2; // it deallocates the memory of object a1  
  }  
}
```



17) Explain the substring(),concat(),replace() methods of class string.(6 Marks)

Ans Refer 9 ans

ii) String concat(String str)

This method creates a new object that contains the invoking string with the contents of str appended to the end. concat() performs the same function as +.

For example,

```
String s1 = "one";
```

```
String s2 = s1.concat("two");
```

puts the string —onetwo into s2

**18) Write a program to implement vector class and its method for adding and removing elements.
(6 Marks)**

Ans

```
import java.util.*;
```

```
public class VectorDemo
```

```
{
```

```
public static void main(String args[])
```

```
{
```

```
Vector v = new Vector();
```

```
v.addElement(new Integer(1));
```

```
v.addElement(new Double(2523));
```

```
v.addElement(new Float(3.55f));
```

```
v.addElement("java");
```

```
System.out.println("Size of vector after four additions: " + v.size());
```

```
v.removeElementAt(2);
```

```
v.removeElementAt(1);
```

```
System.out.println("Size of vector after removed : " + v.size());
```

```
System.out.println("\nElements in vector:");
```

```
System.out.println(v);
```

```
}
```

```
}
```

o/p

Size of vector after four additions:4

Size of vector after removed :2

1 java

19) Explain method overloading with example.(4 Marks)

Ans Refer Q 13 Ans

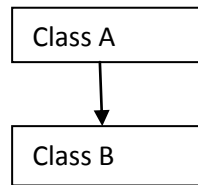
Summer 13

20) What is single inheritance ? Explain with suitable program.(4 Marks)

Ans

The mechanism of creating new classes from old one is called as inheritance. The old class is called known as the base class or super class or parent class, & the new one is called as derived or sub or child class.

The inheritance allows subclasses to inherit all the variables of their parent classes. Inheritance can be of different types. Single level inheritance is that in which there is only one super and sub class.



Example :

//Single level inheritance

```
import java.lang.*;
```

```
class Square //parent or base class
```

```
{
```

```
int length;
```

```
Square(int x)
```

```
{
```

```
length=x;
```

```
}
```

```
void area()
```

```
{
```

```
int area=length*length;
```

```
System.out.println("Area of Square"+area);
```

```
}
```

```
}
```

```
class Rectangle extends Square // child or derived class
```

```
{
```

```
int breadth;
```

```
Rectangl(int x, int y)
```

```
{
```

```
super(x);
```

```
breadth=y;
```

```
}
```

```
void rectarea()
```

```
{
```

```
int area1=length*breadth;
```

```
System.out.println("Area of Rectangle :"+area1);
```

```
}
```

```
}
```

```
class Shape1
```

```
{
```

```
public static void main(String args[])
```

```
{
```

```

Rectangle r = new Rectangle(20,30);
r.rectarea();
r.area();
}
}

```

/* **Output**

```

C:\jdk1.3\bin>java Shape1
Area of Rectangle :600
Area of Square400
*/

```

21) What is difference between arrays and vector? Explain with suitable example. (8 Marks)

Ans Refer 2 Ans

Example 1

```

// Array
class arraytest
{
public static void main(String args[])
{
int arr[]=new int[10];
//storing 1 to 10 numbers in array
for(int i=0;i<10;i++)
{
arr[i]=i+1;
}
//printing elements of array
System.out.println("Array elements :");
for(int i=0;i<10;i++)
{
System.out.print(arr[i]+" ");
}
}
}

```

Example 2 :

```

//Vector
import java.util.*;
class vectortest
{
public static void main(String args[])
{
Vector v=new Vector();
Integer p=new Integer(0);
int l=args.length;
for(int i=0;i<l;i++)
{

```

```

p=Integer.valueOf(args[i]);
v.addElement(p);
}
int size=v.size();
System.out.println("Elements in vector are:");
for(int i=0;i<size;i++)
{
System.out.println(v.elementAt(i));
}
}
}
}

```

22) What are Wrapper classes? Give following wrapper class methods with syntax and use. (6 Marks)

- a) convert integer number to string
- b) convert numeric string to integer number.

Ans. Refer 1 Ans

23) Describe the use of final with respect to inheritance. (4 Marks)

Ans

One use of final keyword is to create named/symbolic constants. Two more uses of final can be seen w.r.t inheritance as:

- 1. Prevent overriding of method**
- 2. Prevent inheritance**

1. Prevent overriding of method:

To disallow a method to be overridden final can be used as a modifier at the start of declaration. Methods written as final cannot be overridden.

e.g.

```

public class A
{
final void mymethod(){...};
}

```

```

public class B extends A
{
final void mymethod(){...}; //forbidden will give error.
}

```

2. Prevent inheritance of over classes:

Final can be used to even disallow the inheritance, to do this a class can be defined with final modifier, declaring a class as final declares all its method as final

e.g.

```

final class test
{
void disp()

```

```

{
Sop ("In superclass");
}
}
Class test 1 extends test // error as class test is final
{
...
}

```

Winter 12

24) What is final variable and method ?How is it different from abstract method.(6 Marks)

Ans Refer 10 ans

25) Explain overriding.How is it different from overloading?Give Example.(6 Marks)

Ans

Overloading

In Java, it is possible to create methods that have the same name, but different parameter lists and different definitions. This is called as method Overloading. Method Overloading is used when objects are required to perform similar tasks but using different input parameters. When we call a method in an object, java matches up the method name first and then number and type of parameters to decide which one of the definitions to execute. This process is known as compile time polymorphism. Return types can also be different.

Example:

Class room

```

{
float length;
float breath;
Room(float x, float y)
{
length =x;
breath=y;
}
Room (float x)
{
length =breath=x;
}
int area()
{
return (length*breath);
}
}

```

Overriding

There may be occasions when we want an object to respond to the same method but have different behavior when that method is called. This is possible by defining a method in the subclass that has the same name, same arguments and same return type as a method in the superclass. Then, when

that method is called, the method defined in the subclass is invoked and executed instead of the one in the superclass. This is known as runtime polymorphism overriding.

```
class Super
{
int x;
Super (int x)
{
this.x=x;
}
void display()
{
System.out.println("Super x= +x");
}
}
Class Sub extends Super
{
int y;
Sub( int x, int y)
{
super (x);
this.y=y;
}
void display()
{
System.out.println("Super x= +x");
System.out.println("Sub y= +y");
}
}
class overrideTest
{
public static void main(String arg[])
{
Sub s1= new Sub(100,200);
s1.display();
}
}
```

26) What is difference between String class and StringBuffer class When it is used.(4 Marks)

Ans Refer Ans 4

String: It is the most commonly used class which can be used to store string constants. String constants can be any characters from keyboard enclosed in double quotes. Strings can be used when data need not be altered.

StringBuffer: It is a peer class of string which can be modified in terms of length & contents.

27) Explain the declaration and use of vector using proper example.(4 Marks)

Ans

Use of Vector:- Vector class is in java.util package.This class can be used to create a generic dynamic array known as Vector that can hold objects of any type & number.It grows automatically.The objects do not have to be homogeneous.Arrays can be easily implemented as Vectors.Vector has dynamic memory allocation and no wastage of memory.Vector Provides methods for adding and removing elements.

Declaration of vector object:-

Vector v1=new Vector(); ☐ without size

Vector v1=new Vector(3); ☐ with size

Example:-

//program to add 5 elements to a Vector and display them.

```
import java.util.*;
```

```
class test1
```

```
{
```

```
public static void main(String args[])
```

```
{
```

```
Vector v=new Vector();
```

```
int l=args.length;
```

```
for(int i=0;i<l;i++)
```

```
{
```

```
v.addElement(args[i]);
```

```
}
```

```
int size=v.size();
```

```
System.out.println("Elements in vector are:");
```

```
for(int i=0;i<size;i++)
```

```
{
```

```
System.out.println(v.elementAt(i));
```

```
}
```

```
}
```

```
}
```

Summer 12

28) List various wrapper classes. Give any four methods of Integer wrapper class.(6 Marks)

Ans

Primitive Data type	Wrapper class
boolean	Boolean
int	Integer
char	Character
float	Float
boolean	Boolean

Methods of Integer wrapper class (Any four)

1. static Integer valueOf(String str) throws NumberFormatException

2. static Int parseInt(String str) throws NumberFormatException
3. static String toString(int num)
4. static double doubleValue()
5. static int intValue()
6. static float floatValue()

29) What is difference between overloading and overriding methods ?(8 Marks)

Ans

Sr. No.	overloading	overriding
1.	It happens within the same class	It happens between super class and subclass
2.	Method signature should not be same	Method signature should be same.
3.	Method can have any return type	Method return type should be same as super class method
4.	Method can have any access level	Method must have same or wide access level than super class method access level
5.	Overloading is early binding or static binding	Overriding is late binding or dynamic binding
6.	They have the same name but, have different parameter lists, and can have different return types.	They have the same name as a superclass method. They have the same parameter list as a superclass method. They have the same return type as a superclass method
7.	It is resolved at compile time	It is resolved at runtime.
8.	For e.g. Refer Ans 25	For e.g. Refer Ans 25

30) What is difference between array and vector ? Explain element at () and add element () method.(6 Marks)

Ans Refer 2 ans 14

31) What is constructor ? Demonstrate the use of parameterized constructor with a suitable example.(4 Marks)

Ans Refer 7

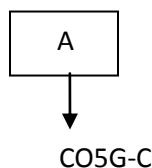
32) What are different form of inheritance?

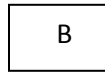
Ans

The mechanism of creating a new class from an old one is called as inheritance. Here the new class can acquire the properties inherited from old class. The old class is known as base class or super class or parent class & the new one is called as derived class or subclass or child class.

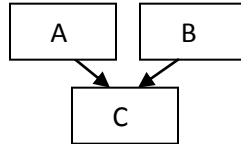
inheritance can be of different types:

1. **A single level inheritance** – where there can be one parent or super class to one subclass



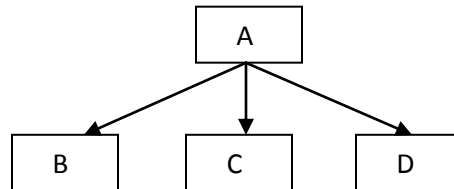


2. **Multiple inheritance**- where there can be multiple parent/super classes to a derived class.



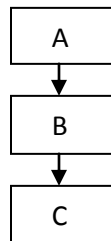
Where A & B are of super classes & C is a subclass. In java, only with the help of classes multiple inheritance is not possible. You require another concept called as 'interface' to achieve multiple inheritance.

3. **Hierarchical inheritance**- where there can be one super class & many classes.



Where A is super class & B, C, & D are subclasses.

4. **Multilevel inheritance** – where there can be derived class derived from another sub class.



Where 'C' sub class is derived from subclass 'B' which is derived from super class 'A'.

Syntax to implement inheritance in java is

class subclass extends super class.

```
{
variable declaration;
method declaration;
:
}
```

e.g. class fruit //base class

```
{
:
}
```

class mango extends fruit // derived class

```
{
```

```
:  
}
```

Important Programs

Summer-15

33) Define a class 'employee' with data members empid, name and salary. Accept data for five objects using Array of objects and print it. (6 Marks)

Ans

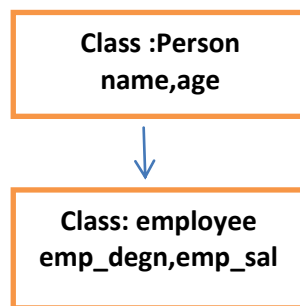
```
import java.lang.*;  
class Employee  
{  
String name;  
int empid;  
int salary;  
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  
void get()  
{  
try  
{  
System.out.println("Enter the Name:");  
name=br.readLine();  
System.out.println("Enter the empid:");  
empid=Integer.parseInt(br.readLine());  
System.out.println("Enter the salary:");  
salary=Integer.parseInt(br.readLine());  
}  
catch(IOException e)  
{  
}  
}  
void show()  
{  
System.out.println("Name:"+name);  
System.out.println("EmpId:"+empid);  
System.out.println("Salary:"+salary);  
}  
}  
class emp  
{  
public static void main(String args[])  
{  
Employee e[]=new Employee[5];  
for(int i=0;i<5;i++)  
{
```

```

e[i]=new Employee();
}
for(int i=0;i<5;i++)
{
e[i].get();
}
for(int i=0;i<5;i++)
{
e[i].show();
}
}
}
}

```

34) Write a program to implement following inheritance: (4Marks)



Ans

class person

```

{
String name;
int age;
void accept(String n,int a)
{
name=n;
age=a;
}
void display()
{
System.out.println("name--->" +name);
System.out.println("age--->" +age);
}
}

```

class employee extends person

```

{
String emp_designation;
float emp_salary;
void accept_emp(String d,float s)
{
emp_designation=d;
}
}

```

```

emp_salary=s;
}
void emp_dis()
{
System.out.println("emp_designation-->" + emp_designation);
System.out.println("emp_salary-->" + emp_salary);
}
}
class single_demo
{
public static void main(String ar[])
{
employee e=new employee();
e.accept("ramesh",35);
e.display();
e.accept_emp("lecturer",35000.78f);
e.emp_dis();
}
}

```

Summer-14

35) Define a class 'employee' with data membersempid, name and salary. Accept data for five objects using Array of objects and print it. (8 Marks)

Ans Refer 33 Ans

36) Write a program to accept as many integers as user wants in a vector.Delete a specific element and display remaining list.(8 Marks)

Ans

```

import java.util.*;
import java.io.*;
public class VectorDemo
{
public static void main(String args[]) throws IOException
{
Vector v = new Vector();
BufferedReader obj = new BufferedReader (new InputStreamReader(System.in));
System.out.println("Enter number of elements u want to add ");
int n= Integer.parseInt(obj.readLine());
for( int i=0; i< n;i++)
{
System.out.println("Enter Integer:");
Integer p= new Integer(Integer.parseInt(obj.readLine()));
v.addElement(p);
}
System.out.println("Enter index of element to be deleted");

```

```

int n1= Integer.parseInt(obj.readLine());
v.removeElementAt(n1);
System.out.println("\nElements in vector after removing specific element:");
for( i=0 ;i<v.size(); i++)
{
System.out.println(v.elementAt(i));
}
}
}
}

```

37) Write a program to accept number from user & convert it into binary by using wrapper class method.

Ans

```

import java.io.*;
public class MyIntegerToBinary
{
public static void main(String args[])throws IOException
{
int i;
BufferedReader obj = new BufferedReader (new InputStreamReader(System.in));
System.out.print("Enter number that you want to convert to binary: ");
i=Integer.parseInt(obj.readLine());
String binary = Integer.toBinaryString(i); //Converts integer to binary
System.out.println("Binary value: "+binary);
}
}
}

```

Winter 13

38) Define a class 'employee' with data members empid, name and salary. Accept data for five objects using Array of objects and print it. (8 Marks)

Ans Refer Ans 33

Summer 13

39) What is array object? Define a class employee with data members empid, name and salary. accept data for five object using array of objects and print it. (6 Marks)

Ans

array object

When same type of data is to be stored in contiguous memory allocations arrays are used. When objects are stored in array, it is called array of objects.

Refer 33 Ans

Winter 12

40) Write a program to calculate gross salary of 5 employees if Basic, HRA and DA are given. (4 Marks)

Ans

```
import java.lang.*;
class Employee
{
String name;
int empid;
float basal,da,hra,grosssal;
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
void get()
{
try
{
System.out.println("Enter the Name:");
name=br.readLine();
System.out.println("Enter the empid:");
empid=Integer.parseInt(br.readLine());
System.out.println("Enter the basic salary:");
basal =Integer.parseInt(br.readLine());
}
catch(IOException e)
{}
}
void show()
{
da= basal*10/100;
hra=basal*15/100;
grosssal= basal+ da+hra;
System.out.println("Name:"+name);
System.out.println("EmpId:"+empid);
System.out.println("grosssal:"+ grosssal);
}
}
class emp
{
public static void main(String args[])
{
Employee e[]=new Employee[5];
for(int i=0;i<5;i++)
{
e[i]=new Employee();
}
for(int i=0;i<5;i++)
{
e[i].get();
}
}
```

```
for(int i=0;i<5;i++)
{
e[i].show();
}
}
}
```

Summer 12

41) Define a class 'employee' with data members as empid, name and salary. Accept data for 5 objects and print it.(6 Marks)

Ans Refer 33 Ans