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

Chapter 1

Introduction to Java

Important Questions

1) State any four features of java.(4 Marks)

Ans:

1. Compile & Interpreted: Java is a two staged system. It combines both approaches. First java compiler translates source code into byte code instruction. Byte codes are not machine instructions.In the second stage java interpreter generates machine code that can be directly executed by machine. Thus java is both compile and interpreted language.

2. Platform independent and portable: Java programs are portable i.e. it can be easily moved from one computer system to another. Changes in OS, Processor, system resources won‘t force any change in java programs. Java compiler generates byte code instructions that can be implemented on any machine as well as the size of primitive data type is machine independent.

3. Object Oriented: Almost everything in java is in the form of object. All program codes and data reside within objects and classes. Similar to other OOP languages java also has basic OOP properties such as encapsulation, polymorphism, data abstraction, inheritance etc. Java comes with an extensive set of classes (default) in packages.

4. Robust & Secure: Java is a robust in the sense that it provides many safeguards to ensure reliable codes. Java incorporates concept of exception handling which captures errors and eliminates any risk of crashing the system. Java system not only verify all memory access but also ensure that no viruses are communicated with an applet. It does not use pointers by which you can gain access to memory locations without proper authorization.

5. Distributed: It is designed as a distributed language for creating applications on network. It has ability to share both data and program. Java application can open and access remote object on internet as easily as they can do in local system.

6. Multithreaded: It can handle multiple tasks simultaneously. Java makes this possible with the feature of multithreading. This means that we need not wait for the application to finish one task before beginning other.

7. Dynamic and Extensible: Java is capable of dynamically linking new class library‟s method and object. Java program supports function written in other languages such as C, C++ which are called as native methods. Native methods are linked dynamically at run time.

2) Explain : Platform independent, Compiled and interpreted features of java(4 Marks)

Ans:

1) Platform independent

Java programs are platform independent and portable. That is they can be easily moved from one computer system to another. Changes in operating systems,processors, system resources will not force any change in java programs. Java compilers generate byte code instructions that can be implemented on any machine as well as the size of primitive data type is machine independent. In this sense, Java programs are platform independent.

2) Compiled & interpreted features of Java

Java is a two staged system. It combines both approaches. First java compiler translates source code into byte code instructions. In the second stage java interpreter generates machine code that can be directly executed by machine. Thus java is both compiled and interpreted language.

3) Explain,why Java is called as true object oriented language(4 Marks)

Ans:

To be truly considered "object oriented", a programming language should support at a minimum four characteristics. Java supports all of these features.

Objects and Classes: Objects are the basic run-time entities. The entire set of data and the code of an object are defined in a class.

Data Abstraction and Encapsulation:wrapping up of data and methods in a single unit called class.

Polymorphism:means ability to take more than one forms.

Inheritance: It is the process by which objects of one class acquire the properties of objects of another class.

Dynamic binding: The code associated with a given procedure call is not known until the time of the call at run time.

Message Communication: A message for an object is a request for execution of a procedure, and therefore will invoke a method in the receiving object that generates the desired result.

4) Explain 4 components of JDK. (4 Marks)

Ans:

component of JDK.

1. Java Compiler – Java Compiler is used to compile java files. It translate java source file to byte code.Java Compiler component of JDK (Java Development Kit) is accessed using —javac command.

2. Java Interpreter – Java Interpreter is used to interpret the java files that are compiled by Java Compiler.It is used to run java program.Java Interpreter component of JDK (Java Development Kit) is accessed using —java command.

3. Java Disassembler- Java Disassembler is used to convert a bytecode files into a java program. Java Dissembler component of JDK (Java Development Kit) is accessed using ”javap” command.

4. Java Header File Generator - Java Header File Generator is used to generate C language header files and source files to implement the native methods. Java Header File Generator component of JDK (Java Development Kit) is accessed using —javah command.

5. Java Documentation – It is used to create HTML- format documentation from java source code file.javadoc command is used.

6. Java Debugger – Java Debugger is used to debug the java files. Java Debugger component of JDK (Java Development Kit) is accessed using —jdb command.

7. Java Applet Viewer – Java Applet Viewer is used to view the Java Applets. Java Applet Viewer component of JDK (Java Development Kit) is accessed using —appletviewer command.

5) What is bytecode? Explain two tools available in JDK.(4 Marks)

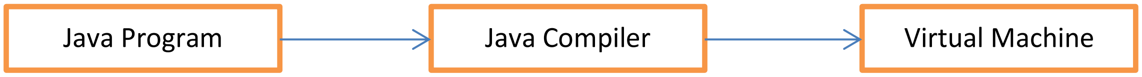
Ans: Refer ans 4 and 6

6) What is bytecode? Explain JVM. (6 Marks)

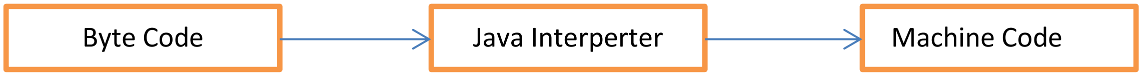
<u>Ans</u>:

<u>JVM is the Java Virtual Machine</u>

All languages compiler translate source code into machine code for a specific computer.The java compiler compiles produces an intermediate code known as byte code for the java virtual machine, which exists only in the computer Memory It is a simulated computer within the computer and does all the major functions of a real computer



Virtual machine code is not machine specific Machine specific code is generated by Java Interpreter by acting as an intermediary between the virtual machine and the real machine. Interpreter is written for each type of machine.



<u>Byte code:</u> Bytecode is the compiled format for Java programs. Once a Java program has been converted to bytecode, it can be transferred across a network and executed by Java Virtual Machine (JVM). A Bytecode file generally has a .class extension.

7) Define JDK.List the tools available in JDK Explain one in details.(4 Marks)

<u>Ans</u>:

<u>Definition</u>: A Java Development Kit (JDK) is a collection of tools which are used for developing, designing, debugging, executing and running java programs.

<u>Tools of JDK.</u>

Java,javap,javah,Javadoc,jdb,appletviewer,javac

<u>1. Java Compiler</u> – it is used to translate java source code to bytecode files that the interpreter can understand.Javac command is used.

8) What do you mean by type Casting?When is it needed.(4 Marks) Or

9) Explain type casing with suitable example.(4 Marks)

<u>Ans</u>: The process of converting one data type to another is called casting. Casting is necessary when method return a type different than the one we require. A boolean value cannot be assigned to any other data type.

There are 2 types of type casting:

<u>1) Implicit Conversion (Widening or Automatic type conversion):</u> Conversion from one primitive datatype to another datatype that has more bits. Widening type conversion will occur only if following two conditions are met.

The two types are compatible.

The destination type is larger than the source type.

e.g int x = 10; // occupies 4 bytes

double y = x; // occupies 8 bytes, no type casting required

System.out.println(y); // prints 10.0

2) Explicit Conversion (Narrowing conversion): Conversion from one primitive data type to another datatype that has fewer bits. Can result into loss of data. floating-point values are truncated when they are cast to integers.

double x = 10.5; // 8 bytes
int y = x; // 4 bytes ; raises compilation error —loss of precision .
Syntax :
Type variable1 = (type) variable2 ;
double x=10.5;
int y=(int)x;
System.out.println(x); //10.5
System.out.println(y): //10
Type casting is used when the conversion is between two incompatible types

10) Write all primitive data types available in java with their storage sizes in bytes. (4 Marks)

Ans.

Java defines eight primitive datatypes as:
byte,short,int,long,float,double,char,Boolean.

Sr. No	Type	Keyword	Size (bit)
1	long integer Integer short integer Byte	long int short byte	64(8 bytes) 32 (4 bytes) 16 (2 bytes) 8 (1 bytes)
2	Double Float	double float	64 (8 bytes) 32 (4 bytes)
3	Character	char	16 (2 bytes)
4	Boolean	boolean	8 (1 bytes)

11) Describe? :(Ternary operator) in Java with suitable example. (4 Marks)

Ans:

1) „?:‟ is called as conditional operator or ternary operator.
2) It can be used to check condition in one line, instead of writing
if…else statement.
3) Its format is (condtion ? true case : false case)
4) example
int a,b;
a=10;
b=(a>5? 12 : 20);
Here b= 12 as a is 10 and condition is true. If value of a is changed to 3
then b will have value as 20.

12) Explain Bit-Wise operator with example. (4 Marks)

Ans:

1) Bitwise NOT (~): called bitwise complement, the unary NOT operator, inverts all of the bits of its operand.

e.g ~ 0111 (decimal 7)

= 1000 (decimal 8)

2) Bitwise AND (&): the AND operator, &, produce a 1 bit if both operands are also 1, A zero is produced in all the cases.

e.g 0101 (decimal 5)
&0011 (decimal 3)
= 0001 (decimal 1)

3) Bitwise OR(|) : the OR operator, | , combines bits such that if either of the bits in the operand is a 1, then the resultant bit is a 1

e.g 0101 (decimal 5)
|0011 (decimal 3)
= 0111 (decimal 7)

4) Bitwise XOR(^) : the XOR operator , ^ , combines bits such that if exactly one operand is 1,then the result is 1. Otherwise result is zero.

e.g 0101 (decimal 5)
^ 0011 (decimal 3)
= 0110 (decimal 6)

5) The Left Shift (<<): the left shift operator, <<, shifts all of the bits in a ‗value' to the left a specified number of times specified by ‗num'

General form : value <<num
e.g. x << 2 (x=12)
0000 1100 << 2
= 0011 0000 (decimal 48)

6) The Right Shift (>>): the right shift operator, >>, shifts all of the bits in a ‗value' to the right a specified number of times specified by ‗num' General form: value >>num.

e.g. x>> 2 (x=32)
0010 0000 >> 2
= 0000 1000 (decimal 8)

7) Unsigned Right Shift (>>>) : >>> always shifts zeros into high order bit.

e.g. int a= -1
a=a>>>24
11111111 11111111 11111111 11111111 (-1 in binary as int) >>> 24
00000000 00000000 00000000 11111111(255 in binary as an int)

8) Bitwise Operators Compound Assignments: All of the above binary bitwise operators have a compound form similar to that of the algebraic operators, which combines the assignment with the bitwise operation. These two statements are equivalent.

e.g. a = a >> 4 ;
a >> = 4;

13) Explain any two logical operators and any two bitwise operators(4 Marks)

Ans:

The logical operators are || , && , !
|| the OR operator
&& the AND operator
! the NOT operator

These operators act on Boolean operands according to this table && is logical and. && combines two boolean values and returns a boolean which is true if and only if both of its operands are true. For instance

```
boolean b;
b = 3 > 2 && 5 < 7; // b is true
b = 2 > 3 && 5 < 7; // b is now false
```

|| is logical or. || combines two boolean variables or expressions and returns a result that is true if either or both of its operands are true. For instance

```
boolean b;
b = 3 > 2 || 5 < 7; // b is true
b = 2 > 3 || 5 < 7; // b is still true
b = 2 > 3 || 5 > 7; // now b is false
```

The last logic operator is ! which means not. It reverses the value of a boolean expression. Thus if b is true !b is false. If b is false !b is true.

```
boolean b;
b = !(3 > 2); // b is false
b = !(2 > 3); // b is true
```

OP1	OP2	OP1||OP2	OP1&&OP2	!OP1
T	T	T	T	F
T	F	T	F	F
F	T	T	F	T
F	F	F	F	T

14) Describe arithmetic operator with Example. (4 Marks)

Ans:

Arithmetic operators are used to construct mathematical expressions
Java provides all the basic arithmetic operators.
The arithmetic operators are:

Operator	Meaning
+	Addition or unary plus
-	Subtraction or unary minus
*	Multiplication
/	Division
%	Modulo division

Integer arithmetic:

When both the operands in a single arithmetic expression are integers, the expression is called integer arithmetic.It always yields an integer value

Real arithmetic:

When the operands of the arithmetic expression are real, the expression is called real arithmetic

Mixed mode arithmetic:

When one of the operands in an expression is real and the other integer, the expression is called mixed mode arithmetic expression.If either operand is real, then the other is also converted to real and real arithmetic is performed.

Example :

```
class ArithmeticOperatorsExample {
public static void main(String args[]) {
int a = 10, b = 5, c;
float d = 4.1f, e = 8.4f,f;
c = a + b;
/*Integer arithmetic*/
System.out.println("The sum of integer numbers a and b is: "+c);
c = a - b;
System.out.println("The difference of integers a and b is: "+c);
c = a * b;
System.out.println("The product of integers a and b is: "+c);
c = a / b;
System.out.println("The quotient of integers a and b is: "+c);
c = a % b;
System.out.println("The remainder of integers a and b is: "+c);
}
}
```

15) Write any four mathematical function(4 Marks)

Ans

1) min() :

Syntax: static int min(int a, int b)

Use: This method returns the smaller of two int values.Parameter can be double ,float,int and long datatype.

2) max() :

Syntax: static int max(int a, int b)

Use: This method returns the greater of two int values. Parameter can be double ,float,int and long datatype

3) sqrt()

Syntax: static double sqrt(double a)

Use:This method returns the correctly rounded positive square root of a double value.Parameter is double.

4) pow() :

Syntax: static double pow(double a, double b)

Use : This method returns the value of the first argument raised to the power of the second argument.

5) exp()

Syntax: static double exp(double a)

Use : This method returns Euler's number e raised to the power of a double value.Parameter is double.

6) round() :

Syntax: static int round(float a)

Use : This method returns the closest int to the argument.

7) abs()

Syntax: static int abs(int a)

Use : This method returns the absolute value of an int value. .Parameter can be double ,float,int and long datatype.

16) With suitable example explain for loop.also give syntax of for loop. (4 Marks)

Ans:

for loop is entry controlled loop that provides concise loop control structure

The general structure of for loop is

```
for (initialization; test condition ; increment)
{
Body of the loop
}
```

- Initialization of the control variable is done first, using assignment statements e.g.:-

int i = 0

- Value of the control variable is tested using the test condition. The test condition is a relational expression, e.g.: i<10, which determine when the loop will exit. If the condition is true the body of the loop is executed else loop is terminated

- After the body of the loop is executed the control transfers back to for statement and the control variable is incremented using statement such as I = i+1

Example:

```
class ForDemo {
public static void main(String a[]) {
System.out.println("Printing the numbers from 1 to 10");
for(int i = 1; i <=10; i++) {
System.out.println(i);
}
}
}
```

The above program is to print numbers from 1 to 10. The loop control variable is set to 1. The number is tested for and printed. After the number is printed, the variable value is incremented by the expression i++. The value of variable is tested and printed. The process is repeated till the value of I reaches 10.

17) Illustrate with example the use of switch case statement. (4 Marks)

Ans:

Java has a built-in multi way decision statement known as switch. The switch case statement is useful when user have multiple choices. Switch statement tests the value of a given variable against a list of case values & when match is found, a block of statements associated with that case is executed.

Syntax:

```
switch(expression)
{
case value1:
block1;
break;
case value2:
block2;
break;
-----
----
default :
Default block;
break;
}
statement n;
```

Example:

```
class switchdemo
{
public static void main ( string args[ ] )
{
int a = Integer.parseInt ( args[0 ] ;
switch (a/5)
{
case 1:
system.out.println(“Poor”);
break;
case 2:
system.out.println(“work Hard”);
break;
case 3:
system.out.println(“Good”);
break;
case 4:
system.out.println(“Very Good”);
break;
case 5:
system.out.println(“Excellent”);
break;
default:
system.out.println(“ Invalid value entered”)
}
}
}
```

18) Explain break and continue statement with example. (4 Marks)

Ans:

Break:

The break keyword is used to stop the entire loop. The break keyword must be used inside any loop or a switch statement.The break keyword will stop the execution of the innermost loop and start executing the next line of code after the block.

Example:

```
public class Test
{
public static void main(String args[])
{

int x[] = {10, 20, 30, 40, 50};
for(int i=0; i<5; i++ )
{
if( x[i] = = 30 )
break;
System.out.print( x[i] );
System.out.print("\n");
}
}
}
```

Continue:

The continue statement skips the current iteration of a for, while , or do-while loop. The unlabeled form skips the following statement and continue with the next iteration.A labeled continue statement skips the current iteration of an outer loop marked with the given label.

Example:

```
public class Test {
public static void main(String args[]) {
int x[] = {10, 20, 30, 40, 50};
for(int i=0; i<5; i++ )
{
if( x[i] = = 30 )
continue;
System.out.print( x[i] );
System.out.print("\n");
}
}
}
```

Remaning Important Questions

19) Explain relational and logical operator in Java. (4 Marks)

Ans. Refer Ques 4 Ans

Relational operators

Java supports 6 relational operators.The value of relational is true or false. It compare 2 quantities.

Operator	Meaning	Example	Eg Result
<	Less than	45<10	False
>	Greater than	40>25	True
<=	Less than or equal to	20<=30	True
>=	Greater than or equal to	4.6>=0.89	True
!=	Not equal to	10!=9	True

20) State any four decision making statement along with their syntax.(4 Marks)

21) Why Java is not 100% object oriented language. (4 Marks)

Ans

Java is not 100% object oriented programming languages because of following reasons.

- **Primitive datatypes (int,float,char etc) are not an object or class.**so java provides wrapper class for these primitive types to make them into objects types.
- **Multiple inheritance through classes is not possible.**Instead of that we can use interface.With the help of interface we can implement multiple inheritance.
- **Availability of static methods & variables:**It allows programmer to invoke a method without instantiating any object breaking rules of encapsulation.
- **Operator overloading is not possible in java:** + operator can be used in 2 different operator (addition operator & concatenation of String) .There is no possibilities in java to do operator overloading.Operator overloading is a kind of polymorphism.

22) Explain: Platform independence, Data Encapsulation features of java. (4 Marks)

Ans Refer Ques 2 Ans

Data Encapsulation

Which means data and function are wrapped into a single unit called class.The data which is declared as private cannot directly accessed to the outside of the class and only the function which are wrapped in the class can access it. Therefore, this function provide interface between the object data and the program.

Remaining Questions

23) Why Java is popular for internet? Explain. (4 Marks)

Important Programs

1) Write a Program to generate Fibonacci series. (4 Marks)

Ans:

```
Class Fibonacci
{
public static void main(String args[])
```

```
{
int f1=1,f2=1.f3,n=1;
System.out.print(f1 + “ ”+ f2 );
While(n <10)
{
f3=f1+f2;
System.out.println(“   “+f3);
f1 =f2;
f2 =f3;
n++;
}
}
```

Using for loop

```
 Class Fibonacci
{
public static void main(String args[])
{
int num=Integer.parseInt(args[0]);
int f1=,f2=0,f3=1;
for(int i=1;i<num;i++)
{
System.out.println(“ “ + f3 + “ “);
f1=f2;
f2=f3;
f3=f1+f2;
} } }
```

<u>o/p</u>

```
1 1 2 1 3 5 8 13 21 34 55 89
```

2) Write a program whether the entered number is prime or not. (4 Marks)

Ans:

```
Class  prime
{
public static void main(String args[])
{
int  num=Integer.parseInt(args[0])
int flag= 0;
for(int i=2;i<num;i++)
{
If(num%i==0)
{
System.out.println(num+" is not a prime ");
flag=1;
break;
}
}
If(flag==0)
System.out.println(num+”is a prime number”);
}
}
```

o/p

```
javac prime.java
java prime 5
5 is a prime number
java prime 6
5 is not a prime number
```

3) Write a program to print sum of numbers. (4 Marks)

Ans:

```
class  SumOfDigits
{
public static void main(String args[])
{
int n = Integer.parseInt(args[0]); //Student may have assumed any other number
int sum = 0, n1;
while (n > 0)
{
n1 = n%10;
n = n/10;
sum = sum+n1;
}
System.out.println("Number is: ”+n);
System.out.println("The sum of the digits of the number is :"+sum);
}
}
```

o/p

```
javac SumOfDigits 321
Number is : 321
The sum of the digits of the number is :6
```

4) Write a Program to calculate percentage of 4 subjects and print in suitable format. (4 Marks)

Ans:

```
import java.lang.*
Import java.io.*;
Class percentage
{
public static void main(String args[])
{
```

```
float sub1,sub2,sub3,sub4,per
DataInputStream  in = new   DataInputStream (System.in);
try
{
System.out.println(”Enter marks of subject 1:”);
sub1 = Flloat.parseFloat(in.readLine());
System.out.println(”Enter marks of subject 2:”);
sub2 = Flloat.parseFloat(in.readLine());
System.out.println(”Enter marks of subject 3:”);
sub3 = Flloat.parseFloat(in.readLine());
System.out.println(”Enter marks of subject 4:”);
sub4 = Flloat.parseFloat(in.readLine());
per = (sub1+sub2+sub3+sub4)/4;
System.out.println(”Percentage=”+per);
}
Catch(Exception e)
{
System.out.println(“Error”);
}
}
}
```

o/p

```
Enter marks of subject 1: 60.00
Enter marks of subject 2: 50.00
Enter marks of subject 3: 60.00
Enter marks of subject 4: 50.00
Percentage=55.00
```

5) Write a program to accept two numbers as command line arguments and print the addition of these numbers. (4 Marks)

Ans:

```
class commandaddition
{
public static void main(String args[])
{
int num1 = Integer.parseInt(args[0]); //take argument as command line
int num2 = Integer.parseInt(args[0]); //take argument as command line
int Addition = num1 + num2;
System.out.println("Addition of number is : "+Addition);
}
}
```

o/p

```
Javac  commandaddition
Java   commandaddition 10 20
Addition of number is :30
```

6) **Write a program to accept a number as command line argument and print number is even or odd. (4 Marks)**

Ans:

```
public class oddeven
{
public static void main(String key[])
{
int x=Integer.parseInt(key[0]);
if (x%2 ==0)
{
System.out.println("Even Number");
}
else
{
System.out.println("Odd Number");
}}}
```

o/p

```
java oddeven 5
Odd Number
java oddeven 10
Even Number
```

7) **Write a program to print reverse of a number. (4 Marks)**

Ans:

```
class reverse
{
public static void main(String args[])
{
Int num,temp;
num = Integer.parseInt(args[0]);
System.out.println(“Number is:”+num);
System.out.println(“Reverse number is:”+);
do
{
  temp = num%10;
  System.out.print(temp);
  num=num/10;
}
While(num!=0);
}
}
```

o/p

```
javac reverse.java
```

```
java reverse 321
Number is:321
Reverse number is:123
```

8) Write a program to accept a number and print factorial. (4 Marks)

Ans:

```
import java.io.*;
class Factorial
{
public static void main(String args[])
{
String str;
int no,fact;
fact=1;
try{
BufferedReader obj = new BufferedReader(new InputStreamReader(System.in));
System.out.print("Enter number whose Factorial is to be found : ");
str=obj.readLine();
no=Integer.parseInt(str);
while(no > 0)
{
fact=fact*no;
no=no-1;
}
System.out.println("FACTORIAL of a given number is : "+fact);
}
catch(Exception e)
{}
}
}
```

o/p

```
Enter number whose Factorial is to be found : 4
FACTORIAL of a given number is : 24
```

9) Write a program to accept first name, middle name and surname in three different strings and then concatenate the concatenate the three strings to make full name. (4 Marks)

Ans:

```
import java.io.*;
class StringConcat
{
public static void main(String[] args) throws IOException
{
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
String str1,str2,str3,namestr="";
System.out.println("Enter first name:");
str1=br.readLine();
System.out.println("Enter middle name:");
str2=br.readLine();
System.out.println("Enter last name:");
str3=br.readLine();
namestr = namestr.concat(str1);
namestr = namestr + " ";
namestr = namestr.concat(str2);
namestr = namestr + " ";
namestr = namestr.concat(str3);
System.out.print("Your FULL NAME is : " + namestr);
}
}
```

o/p

```
java StringConcat
Enter first name:
Anup
Enter middle name:
Mohit
Enter last name:
Verma
Your FULL NAME is : Anup Mohit Verma
```