Q1)What will be the output?

```
public class Test1{
   public static void main(String[] args){
      int arr[] = new int[3];
      for(int i = 0;i < arr.length;i++){
         System.out.println(arr[i]);
      }
   }
}
```

A1 0 0 0
A2 ArrayIndexoutOfBoundsException
A3 NullPointerException
A4 null null null

Q2)Which of the following are valid array declarations?

A1 int arr[] = new int[];
A2 float arr[10] = new fl
A3 double []arr = new double[10];
A4 None Of the Above.

Q3)What will be the output?

```
public class Test3{
   public void method(){
       System.out.println("Called");
   }
   public static void main(String[] args){
       method();
   }
}
```

A1 "Called"
A2 Compiler
A3 Runtime
A4 Nothing

Q4)What will be the output?

```
public class Test4{
     public static void method(){
         System.out.println("Called");
     }
     public static void main(String[] args){
         Test4 t4 = null;
         t4.method();
     }
```

```
  }
```

A1 "Called"
A2 Compiler
A3 Runtime Exception
A4 Nothing is printed in screen

Q5)What will be the output?

```
public class Test5{
    public void Test5(){
       System.out.println("Constructor1");
    }
    public Test5(){
       System.out.println("Constructor2");
    }
    public static void main(String[] args){
       Test5 t5 = new Test5();
    }
}
```

A1 "Constructor1"
A2 "Constructor2"
A3 "Constructor1""Constructor2"
A4 Compiler Errror

Q6)What will be the output?

```
public class Test6{
    public Test6(){
       this(4);
    }
    public Test6(byte var){
       System.out.println(var);
    }
    public static void main(String[] args){
       Test6 t6 = new Test6();
    }
}
```

A1 4
A2 4 4
A3 Compiler Error
A4 Compiles and Runs without error

Q7)What will be the output?

```
public class Test7{
     public Test7(){}
     public Test7(Test7 ref){
```

```
        this (ref,"Hai");
    }
    public Test7(Test7 ref,String str){
        ref.Test7(str);
        System.out.println("Hi");
    }
    public void Test7(String str){
        System.out.println(str);
    }
    public static void main(String[] args){
        Test7 t = new Test7();
        Test7 t7 = new Test7(t);
    }
}
```

A1 HI
A2 hai
A3 Hai Hi
A4 Hi Hai

Q8)Which of the following are valid Constructors?

A1 public Test8(){}
A2 private void Test8(){}
A3 protected Test8(int k){}
A4 Test8(){}

Q9)Which of the following are valid method declarations?

A1 abstract method(){}
A2 abstract void method(){}
A3 final void method(){}
A4 int method(){}

Q10)Which of the following are valid top-level class declarations?

A1 class Test10
A2 public class Test10
A3 final class Test10
A4 abstract final class Test10

Q11)transient keywaord can be used with?

A1 method
A2 variable
A3 class
A4 constructor

Q12)which of the following are valid combinations for class declaration?

A1 abstract final class Test12{}
A2 abstract static class Test12{}
A3 final static class Test12{}
A4 public final strictfp class Test12{}

Q13)which of the following are valid constructor signatures?

A1 public void className()
A2 public void className()
A3 private className()
A4 static className()

Q14)Which of the following modifiers can be used with top class declaration?

A1 static
A2 private
A3 public
A4 final
A5 abstract

Q15)Which of the following are valid array declarations?

A1 int arr[] = new int[];
A2 int arr[][] = new int [10][10];
A3 float arr[][] = new float[][10];
A4 float arr[] = new float[10];

Q16)What will be the output of the following program?

```
public class Test1 {
      static{
           System.out.println("Static");
      }
      {
           System.out.println("Instance");
       }
       public void Test1(){
           System.out.println("Constructor");
       }
       public static void main(String[] args) {
          Test1 t = null;
       }
}
```

A1 Instance Static
A2 Static Instance
A3 Static
A4 Static Instance Constructor

Q17)What will be the output of the following program?

```
class Sup{
    public Sup(String str){
        System.out.println("Super class");
    }
}

public class Test2 extends Sup{
      public Test2(){
            System.out.println("Sub class");
      }
      public static void main(String[] args) {
            Test2 t2 = new Test2();
      }
}
```

A1 Super class,SubClass
A2 Super class
A3 Sub class
A4 Compiler Error

Q18)What will be the output of the following program?

```
public class Test3 {
      public static void main(String[] args) {
            System.out.println("Main Method1");
      }
      public static void main(String args){
            System.out.println("Main Method2");
     }
}
```

A1 Main Method1
A2 Main Method1 Main Method2
A3 Main Method2
A4 Runtime Exception

Q19)What will be the output of the following program?

```
public class Test4 {
     public static void main(String args) {
           System.out.println("Sample Program");
     }
}
```

A1 Sample Program
A2 Compiler Error
A3 Runtime Exception
A4 None

Q20)What will be the output of the following program?

```
class Sup1{
      public Sup1(){
           System.out.println("Hai");
      }
      private Sup1(String str){
          System.out.println(str);
      }
}

public class Test5 extends Sup1{
      private Test5(String str){
           System.out.println(str);
           super();
      }
      public static void main(String[] args) {
          Test5 t5 = new Test5("HI");
      }
}
```

A1 Hai,Hi,Hi
A2 Hai,Hi
A3 Hi,Hi
A4 Compiler Error

Q21)Which of the following are not a wrapper class?

A1 String
A2 Integer
A3 StringBuffer
A4 Boolean

Q22)Select the correct syntax for main method :

A1 public void main(String args[])
A2 public static void main(String args)
A3 public static void Main(String args[])
A4 None of the Above

Q23)Which of the following are not a valid declarations?

A1 float f = 1;
A2 float f = 1.2f;
A3 float f = 1.2;
A4 float f = (float)1.2;

Q24)String s1 = new String("hi");

```
String s2 = "hi";
System.out.println(s1==s2);
System.out.println(s1.equals(s2));
```

A1 false true
A2 true false
A3 true true
A4 None of the above.

Q25)Integer i = new Integer(0);

```
Float f = new Float(0);
System.out.println(i==f);
System.out.println(i.equals(f));
```

A1 true false
A2 false true
A3 true true
A4 Compiler error

Answers

1) A1 is correct
Local Array variables are initialized to their default values.

2) A3)double []arr = new double[10];
A1 is not valid because (new int[]) array size must be specified unless it is annonymous array declaration.
A2 is not valid because array declaration must not specify the size of the array.

3) A2) Compiler Error
non-static(instance) methods or variables are cannot be referenced from static context.Compiler will give the following error while compiling the above code: Test3.java:6: non-static method method() cannot be referenced from a static context method();

4) A1) "Called"
Static methods canbe called only by referance.Because its not binded with objects.So, in this case "null" value doesn't make sense."method" is called without any problem.

5) A2)"Constructor2"
Constructors cannot have return types. If its declared with return types then it is consider as methods. So, output wull be "Constructor2".

6) A3)Compiler Error

7) A3)Hai Hi

8) A1)public Test8(){}
A3)protected Test8(int k){}

9) A3)final void method(){}
A4)int method(){}

10) A1) class Test10
A2) public Test10
A3) final Test10

11) b)variable

12) c)final static class()
d)public final strictfp class{}

13) b)public className()
c)private className()

14) c)public
d)final
e)abstract

15) b)int arr[][] = new int [10][10];
d)float arr[] = new float[10];

16) c)Static

17) d)Compiler Error

18) a)Main Method1

19) c)Runtime Exception

20) d)Compiler Error

21) A1)String
A4)String Buffer

22) A4)None of the Above

23) A3)float f = 1.2;

24)A1) false true

25)A4)Compiler error

~~~~~~

```
import java.util.*;
public class Test1{
    public static void main(String a[]){
        Set s = new TreeSet();
        s.add(new Person(20));
        s.add(new Person(10));
        System.out.println(s);
    }
}
class Person{
   Person(int i){}
}
```

What is the output?

A1 10 20

A2 ClassCastException

A3 Compiler Error

A4 Compiler Error

Q2

```
import java.util.*;
public class Test2{
    public static void main(String a[]){
        Map s = new Hashtable();
        s.put(null,null);
        System.out.println(s);
    }
}
```

What is the output?

A1 [null = null]

A2 NullPointerException

A3 null

A4 []

Q3

```
import java.util.*;
public class Test3{
   public static void main(String a[]){
      Map s = new WeakHashMap(10);
      s.put(null,null);
```
~~~~~~

```
        System.out.println(s);
    }
}
```

What is the output?

A1 [null = null]

A2 NullPointerException

A3 null

A4 []

Q4

```
import java.util.*;
public class Test4{
    public static void main(String a[]){
        Map s = new LinkedHashMap();
        s.put("1","one");
        s.put("3","three");
        s.put("2","two");
        System.out.println(s);
    }
}
```

What is the output?

A1 [1=one,3=three,2=two]

A2 NullPointerException

A3 [1=one,2=two,3=three]

A4 []

Q5

```
import java.util.*;
public class Test5{
    public static void main(String a[]){
        Map s = new HashMap();
        s.put("1","one");
        s.put("3","three");
        s.put("2","two");
        System.out.println(s);
    }
}
```

What is the output?

A1 [1=one,3=three,2=two]

A2 [3=three,2=two,1=one]

A3 cannot predict the order

A4 []

Q6

```
public class Test6{
    public static void method(float f){
        System.out.println("Float");
    }
```

```
    public static void method(double f){
       System.out.println("Double");
    }
    public static void main(String a[]){
       float f1 = 2.0f;
       float f2 = 2.0f;
       method(1.0);
       method(1.0f);
       method(1.0f*2.0f);
       method(1.0f*2.0);
       method(f1*f2);
    }
}
```

What is the output?

A1 Double Double Double Double Double
A2 Float Float Float Float Float
A3 Double Float Float Double Float
A4 Double Float Float Double Double

Q7

```
public class Test7{
   public static void method(byte b){
      System.out.println("Byte");
   }
   public static void method(int i){
      System.out.println("Int");
   }
   public static void main(String a[]){
      byte b = 1;
      method(1);
      method(128);
      method((byte)128);
      method(b);
   }
}
```

What is the output?

A1 Byte Int Int Byte
A2 Byte Int Int Byte
A3 Byte Int Byte Byte
A4 Int Int Byte Byte

Q8

```
public class Test8{
    public static void main(String a[]){
       byte b = 1;
       char c = 2;
       short s = 3;
       int i = 4;
```

```
        c = b; // 1  (Error)
        s = b; // 2
        i = b; //3
        s = c * b; //4
    }
}
```

Which of the following are correct?

A1 Error at mark 1

A2 Error at mark 2

A3 Error at mark 3

A4 Error at mark 4

Q9

```
public class Test9{
public static void main(String a[]){
final byte b = 1;
char c = 2;
short s = 3;
int i = 4;

c = b; // 1
s = b; // 2
i = b; //3
s = c * b; //4
}
}
```

Which of the following are correct?

A1 Error at mark 1

A2 Error at mark 2

A3 Error at mark 3

A4 Error at mark 4

Q10

```
public class Test10{
     public static void main(String a[]){
        String s1 = "Sun";
        String s2 = "Java";
        s1.concat(s2);
        System.out.println(s1);
     }
}
```

What is output?

A1 Sun

A2 Java
A3 SunJava
A4 JavaSun

Q11

```
public class Test11{
    public static void main(String a[]){
        Integer i1 = new Integer(127);
        Integer i2 = new Integer(127);
        Long l = new Long(127);
        System.out.println(i1 == i2);
        System.out.println(i1.equals(i2));
        System.out.println(i1.equals(l));
    }
}
```

What is output?

A1 false true true
A2 true true true
A3 false true false
A4 Compiler Error

Q12

```
public class Test12{
   public static void main(String a[]){
       byte b = 100;
       Byte b1= new Byte(100);
       Byte b2 = new Byte(b);
       System.out.println(b1 == b2);
       System.out.println(b1.equals(b2));
   }
}
```

What is output?

A1 false true
A2 true false
A3 true true
A4 Compiler Error

Q13

```
public class Test13{
   public static void method(String s){
      System.out.println("String Version");
   }
   public static void method(StringBuffer sb){
      System.out.println("String Buffer Version");
```

```
    }
    public static void main(String a[]){
       method(null);
    }
}
What is output?
```

A1 String Version
A2 String Buffer Version
A3 Runtime Exception
A4 Compiler Error

Q14

```
public class Test14{
    static String s ="Instance";
    public static void method(String s){
        s+="Add";
    }
    public static void main(String a[]){
        Test14 t = new Test14();
        s = "New Instance";
        String s = "Local";
        method(s);
        System.out.println(s);
        System.out.println(t.s);
    }
}
What is output?
```

A1 Local Instance
A2 Local New Instance
A3 Loca Add New Instance
A4 Compiler Error

Q15

```
public class Test15{
    public static void method(StringBuffer sb){
        sb.append(" Added");
        sb = new StringBuffer("Hai");
    }
    public static void main(String a[]){
        StringBuffer sb = new StringBuffer("String
Buffer");
        method(sb);
        System.out.println(sb);
    }
}
```

What is output?

A1 String Buffer
A2 String Buffer Added
A3 Hai
A4 Compiler Error

Q16
```
public class Test16{
    public static void method(StringBuffer sb){
        sb = new StringBuffer("Hai");
        sb.append(" Added");
    }
    public static void main(String a[]){
        StringBuffer sb = new StringBuffer("String
Buffer");
        method(sb);
        System.out.println(sb);
    }
}
```
What is output?

A1 String Buffer
A2 String Buffer Added
A3 Hai
A4 Compiler Error

Q17
```
import java.util.*;
public class Test17{
    public static void main(String a[]){
        Map m = new Hashtable(10,0.75f);
        System.out.println(m.size());
    }
}
```
What is output?

A1 0

A2 10
A3 7
A4 cOMPILER eRROR

Q18 What is the default capacity of java.util.Hashtable?
A1 10
A2 16
A3 11
A4 20

Q19 What is the default capacity of java.util.HashMap?
A1 10
A2 16
A3 11
A4 20

Q20 Which of the following classes has synchronized methods?
A1 ArrayList
A2 Vector
A3 HashTable
A4 WeakHashMap

Q21
```
public class Test21{
    public static void main(String a[]){
       String s1 = new String("Hai");
       String s2 = "Hai";
       String s3 = new String("Hai").intern();
       System.out.println(s1 == s2);
       System.out.println(s1 == s3);
       System.out.println(s2 == s3);
    }
}
```
What is output?
A1 false false true
A2 true false true
A3 false false false
A4 false true true

Q22
```
public class Test22{
    public static void main(String a[]){
       String s1 = "SunMicroSystems";
       System.out.println(s1.substring(0));
       System.out.println(s1.substring(1,4));
```

```
            System.out.println(s1.substring(8));
        }
    }
    What is output?
```

A1 SunMicrosystems sun oSystem
A2 SunMicrosystems unM Systems
A3 StringIndexOutOfBoundsException
A4 None Of the above

Q23
```
    public class Test23{
        public static void main(String a[]){
            String s1 = "Sun";
            System.out.println(s1.substring(5));
        }
    }
    What is output?
```

A1 -1
A2 0
A3 StringIndexOutOfBoundsException
A4 ArrayIndexOutOfBoundsException

Q24 Which of the following are static methods in java.lang.String class?

A1 valueOf
A2 length
A3 indexOf
A4 All the above.

Q25
```
    public class Test25{
       public static void main(String a[]){
          StringBuffer sb = new StringBuffer(8);
          sb.append("TestString");
          System.out.println(sb.capacity());
          System.out.println(sb.length());
       }
    }

    What is output?
```

A1 8 10
A2 10 10
A3 18 10

A4 18 18

Answers

1	ClassCastException
2	NullPointerException
3	[null = null]
4	[1=one,3=three,2=two]
5	cannot predict the order.
6	Double Float Float Double Float
7	Int Int Byte Byte
8	Error at mark 1 Error at mark 4
9	Error at mark 4
10	Sun
11	false true false
12	Compiler Error
13	Compiler Error
14	Local New Instance
15	String Buffer Added
16	String Buffer
17	0
18	11
19	16
20	Vector HashTable
21	false false true
22	SunMicrosystems unM Systems
23	StringIndexOutOfBoundsException
24	valueOf
25	18 10

Mock Exam - 3

Q1

```
public class Test1{
    public static void main(String args[]){
        System.out.println(method());
    }
    public static int method(){
       try{
          return 1;
       }
       catch(Exception e){
```

```
            return 2;
         }
         finally{
            return 3;
         }
      }
   }
```

What will be the output?

A1 1

A2 2

A3 3

A4 0

Q2

```
   public class Test2{
      public static void main(String args[]){
          System.out.println(method());
      }
      public static int method(){
          try{
              throw new Exception();
              return 1;
          }
          catch(Exception e){
              return 2;
          }
          finally{
              return 3;
          }
      }
   }
```

What will be the output?

1

A1 2

A2 3

A3 4

A4 Compiler error

Q3

```
   public class Test3{
       public static void main(String args[]){
           System.out.println(method());
       }
       public static int method(){
           try{
               throw new Exception();
           }
```

```
            catch(Exception e){
                throw new Exception();
            }
            finally{
                return 3;
            }
        }
    }
```

What will be the output?

A1 3

A2 0

A3 Runtime Exception

A4 Compiler error

```
    public class Test4{
        public static void main(String args[]){
            System.out.println(method());
        }
Q4      public static int method(){
            return;
        }
    }
```

What will be the output?

A1 null

A2 0

A3 Runtime exception

A4 Compiler error

```
    import java.io.IOException;
    public class Test5{
       public static void main(String args[]){
          try{
              throw new IOException();
          }
          catch(Exception e){
Q5            System.out.println("Excepion");
          }
          catch(IOException e){
              System.out.println("IOExcepion");
          }
       }
    }
```

What will be the output?

A1 Exception

A2 IOException

A3 Exception IOException

A4 Compilers error

Q6

```
public class Test6{
    public static void main(String args[]) throws
Exception{
        try{
            throw new Exception();
        }
        finally{
            System.out.println("No Error");
        }
     }
}
```

What will be the output?

A1 No Error followed by java.lang.Exception

A2 java.lang.Exception followed by No Error

A3 No Error

A4 Compiler Error

Q7

```
ublic class Test7{
    public static void main(String args[]) throws
Exception{
        Test7 t = new Test7();
        t.method();
        System.out.println("Print");
    }
    public void method()throws Exception{
        throw new Exception();
    }
}
```

What will be the output?

A1 Print

A2 Exception thrown at runtime

A3 no output

A4 Compiler Error

Q8

```
public class Test8{
    public static void main(String args[]) throws
Exception{
        Test8 t = new Test8();
        t.method();
        System.out.println("Print");
    }
    public void method(){
        try{
            throw new Exception();
```

```
            }catch(Exception e){}
        }
    }
```

What will be the output?

A1 Print

A2 Exception thrown at runtime

A3 no output

A4 Compiler Error

Q9

```
public class Test9 extends A{
    public static void main(String args[]) throws
Exception{
        Test9 t = new Test9();
    }
}
class A{
    A() throws Exception{
        System.out.println("A Class");
    }
}
```

What will be the output?

A1 A Class

A2 Runtimxception

A3 no output

A4 Compiler Error

Q10

```
public class Test10 extends A{
    Test10()throws Exception{
        System.out.println("Test10 Class");
    }
    public static void main(String args[]) throws
Exception{
        Test10 t = new Test10();
    }
}
class A{
    A() throws Exception{
        System.out.println("A Class");
    }
}
```

What will be the output?

A1 A Class Test10 Class

A2 Runtimxception

A3 no output

A4 Compiler Error

Q11

```
public class Test11 extends A{
    Test11()throws Exception{
        System.out.println("Test10 Class");
    }
    Test11(int i){}
    public static void main(String args[]) throws
Exception{
        Test11 t = new Test11();
    }
}
class A{
    A() throws Exception{
        System.out.println("A Class");
    }
}
```

What will be the output?

A1 A Class Test10 Class

A2 Runtimxception

A3 no output

A4 Compiler Error

Q12

```
import java.io.IOException;
    public class Test12 extends A{
        public void method() throws Exception{
        System.out.println("Subclass");
    }
    public static void main(String args[]) throws
Exception{
        A a = new A();
        a.method();
        a = new Test12();
        a.method();
    }
}
class A{
    public void method() throws IOException{
        System.out.println("Superclass");
    }
}
```

What will be the output?

A1 Subclass Superclass

A2 Runtimxception

A3 Superclass Superclass

A4 Compiler Error

Q13 What are the legal arguments types for switch?

A1 int
A2 byte
A3 char
A4 All the above.

Q14 Which of the following are valif if constructs?
A1 if(2>3){}
A2 if(false){}
A3 if(false){}
A4 if(true)

Q15
```
public class Test15{
    public static void main(String args[]) throws
Exception{
        for (int i = 0;i < 3;i++){
            for (int j = 0;j < 3;j++){
                System.out.print(i);
                System.out.print(j+",");
                break;
            }
        }
    }
}
```
What will be the output?
A1 00,
A2 00,10,20,
A3 000102
A4 None of the above

Q16
```
public class Test16 extends A{
    Test16(){
        System.out.println("Sub");
    }
    public static void main(String args[]) {
        Test16 t = new test16();
```
````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````

# Mock Exam - 4

Q1
```
public class Test1{
    static void method(Object obj){
        System.out.println("Object");
    }
    static void method(String str){
```
```

```
        System.out.println("String");
    }
    public static void main(String args[]){
        method(null);
    }
}
```

What will be the output?

A1 String
A2 Object
A3 null
A4 Compiler Error

---

Q2

```
public class Test2{
    static void method(StringBuffer obj){
        System.out.println("StringBuffer");
    }
    static void method(String str){
        System.out.println("String");
    }
    public static void main(String args[]){
        method(null);
    }
}
```

What will be the output?

A1 String
A2 Object
A3 null
A4 Compiler Error

---

Q3

```
class Test{}
public class Test3{
   static void method(Object obj){
       System.out.println("StringBuffer");
   }
   static void method(String str){
       System.out.println("String");
   }
   static void method(Test t){
       System.out.println("Test");
   }
   public static void main(String args[]){
       method(null);
   }
```

```
}
```

What will be the output?

A1 String

A2 Object

A3 Test

A4 Compiler Error

---

Q4

```
public class Test4{
    public static void main(String args[]){
        I i1 = new A();
        I i2 = new B();
        A a = new A();
        System.out.println(i1 instanceof I);
        System.out.println(i2 instanceof B);
        System.out.println(a instanceof I);
    }
}
interface I{}
class A implements I{}
class B implements I{}
```

What will be the output?

A1 true true true

A2 true false true

A3 true false false

A4 Compiler Error

---

Q5

```
public class Test5{
    public static void main(String args[]){
        System.out.println(I.k);
    }
}
interface I{
    int k;
}
```

What will be the output?

A1 true true true

A2 true false true

A3 true false false

A4 Compiler Error

---

Q6

```
public class Test6 implements I{
    int k = 1;
    public static void main(String args[]){
        System.out.println(k);
    }
```

```
}
interface I{
    int k = 0;
}
```

What will be the output?

A1 0

A2 1

A3 null

A4 Compiler Error

---

Q7

```
public class Test7 implements I{
    int k = 1;
    static{
        k = k * 2;
    }
    {
        k = k * 2;
    }
    public static void main(String args[]){
        Test7 t1 = new Test7();
        Test7 t2 = new Test7();
        System.out.println(t1.k);
        System.out.println(t2.k);
        System.out.println(k);
    }
}
```

What will be the output?

A1 0

A2 1

A3 null

A4 Compiler Error

---

Q8

```
public class Test8{
    static int k = 1;
    static{
       k = k * 2;
    }
    {
       k = k * 2;
    }
    public static void main(String args[]){
       System.out.println(k);
    }
}
```

What will be the output?

A1 1

A2 2

A3 4

A4 Compiler Error

---

Q9

```
public class Test9{
    static int k = 1;

    {
        k = k * 2;
    }
    public static void main(String args[]){
        System.out.println(k);
    }
}
```

What will be the output?

A1 1

A2 2

A3 4

A4 Compiler Error

---

Q10

```
public class Test10{
    final static int k;
    static{
        k = 0;
    }
    public static void main(String args[]){
        System.out.println(k);
    }
}
```

What will be the output?

A1 0

A2 1

A3 null

A4 Compiler Error

Answers

1 String

2 Compiler Error

3 Compiler Error

4 true true true

5 true true true

6 Compiler Error

7 Compiler Error

8 2

9 1
10 0

| | |
|---|---|
| Q1 | **Which of the following are the correct form of documentation comments?** |
| A1 | //some text here |
| A2 | /*some text here*/ |
| A3 | /**some text here*/ |
| A4 | all the above |

| | |
|---|---|
| Q2 | **State the correct formula for minimum/maximum values for integer primitives where no_of_bits is the size of the type in bits.** |
| A1 | 2^(no_of_bits-1) / 2^(no_of_bits-1)+1 |
| A2 | 2^(no_of_bits+1) / 2^(no_of_bits+1)+1 |
| A3 | 2^(no_of_bits-1) / 2^(no_of_bits-1)-1 |
| A4 | all the above |

| | |
|---|---|
| Q3 | **Which of the following initializes boolean primitive?** |
| A1 | Boolean flag=true; |
| A2 | boolean flag=true; |
| A3 | boolean flag=TRUE; |
| A4 | Boolean flag=TRUE; |

| | |
|---|---|
| Q4 | **which of the following is the correct way to define a class that will be in the default package** |
| A1 | package default;<br><br>import java.util.*; |
| A2 | import java.util.*;<br><br>package default; |
| A3 | import java.util.*; |
| A4 | all the above |

| | |
|---|---|
| Q5 | **Which of the following main method in a java application is correct?** |
| A1 | public static void main(String[] args) |
| A2 | public void main(String args[]) |
| A3 | public static void main (string[] args) |
| A4 | final public static void main (String[] args) |
| A5 | static public void main(String x[]) |
| A6 | static void main (string[] args) |
| A7 | a and e only. |
| A8 | g and d |

| | |
|---|---|
| Q6 | **Which of the following is default integer primitive** |
| A1 | short |
| A2 | int |
| A3 | byte |
| A4 | char |
| A5 | long |

| | |
|---|---|
| Q7 | **Which of the following is not a reserved word in java** |
| A1 | **import** |
| A2 | finally |
| A3 | friend |
| A4 | goto |

| | |
|---|---|
| Q8 | **When writing a utility class, someclass, which extends mainclass class and will be used by several other classes in a large project. These other classes will be in different packages.Pick the correct class declaration** |
| A1 | class someclass extends mainclass |
| A2 | protected class someclass extends mainclass |
| A3 | public class someclass extends mainclass |
| A4 | none |

| | |
|---|---|
| Q9 | **Which of the following variable names are invalid?** |
| A1 | example |
| A2 | 2sumup |
| A3 | its4u |

| | |
|---|---|
| A4 | $money |

| | |
|---|---|
| Q10 | **Take a look at the following code:**<br><br>**public static void main (String[] args){**<br><br>**System.out.println(args[1]);**<br><br>**}**<br><br>**The above is compiled and then executed by the following command line.**<br><br>**java test one two three four**<br><br>**choose the correct output** |
| A1 | one |
| A2 | two |
| A3 | three |
| A4 | four |
| A5 | none. |

| Answers | |
|---|---|
| 1 | c |
| 2 | c Substitute no_of_bits = ( 8 for byte , 16 for short, 16 for char, 32 for int, 64 for long, 32 for float, 64 for double).We get (2^7) / (2^7) -1 for int and so on for other types |
| 3 | b primitive boolean keyword is 'boolean' and boolean can be only 'true' or 'false' |
| 4 | c. there is nothing like explicit declaration for default package. The class is added to default package if there is no package statement. |
| 5 | h Valid declaration for the main() method must be public and static, have void as return type and take a single array of String objects as arguments. The order of public and static keywords is irrelevant. Also declaring the method final does not effect the method's potential to be used as a main() method. |
| 6 | b |
| 7 | c . There are no friend functions as in C++. |
| 8 | c |
| 9 | b |
| 10 | b Array index start from 0. So somearray[0] points |

to the first element in the array.

Mock 6

1. **Please select signed integrals**
   A. char, byte, and short
   B. byte, short, int, and long
   C. char, short, and long

2. **Java characters are ...**
   A. ASCII code
   B. Binary code
   C. Unicode
   D. ANSI code

3. **Please select the size of an int type**
   A. 32 bytes
   B. 16 bits
   C. 32 bits
   D. 16 bytes

4. **Select default value of boolean type**
   A. true
   B. false
   C. 0
   D. 1

5. **Consider the following line of code:**
   **char x[]=new char[10];**
   **After execution, what is value of x[0]?**
   A. 0
   B. '\u0000'
   C. null

6. **A package statement MUST exist in all classes**
   A. True
   B. False

7. **Please choose invalid identifiers**
   A. temp
   B. BIGint
   C. 25_length

D. !length

8. **Please select floating point types**

A. byte
B. int
C. double
D. short
E. long
F. float

9. **All operands are evaluated left to right**
A. true
B. false

10. **Consider the following line of code:**
**byte x=64;**
**byte y=5;**
**byte z= (byte)(x*y);**
**After execution what is value of z?**
A. 320
B. 0
C. 645
D. 64

11. **Consider the following line of code:**
**int x=7;**
**int y=4;**
**int z=7/4;**
**After execution what is value of z?**
A. 1.75
B. 0
C. 1
D. 2

12. **Please select the true statement for ! operator**
A. The ! operator inverts the value of a boolean expression
B. The ! operator inverts the bit pattern of an integral expression.
C. Both A and B
D. None of the above

13. **Please select arithmetic operations which**

**can result in AirthmeticException**

A. Multiplication: *
B. Division: /
C. Modulo: %
D. Addition: +
E. Subtraction: -

14. **Please select operators which perform bit shifts of the binary representation of the integral types**
    A. <<
    B. >>
    C. >>>
    D. ~
    E. &
    F. ^
    G. |

15. **A protected method may be overridden by ...**
    A. A private method
    B. A protected method
    C. A public method
    D. All of the above

16. **A public method may not be overridden by ...**
    A. A private method
    B. A protected method
    C. A public method
    D. All of the above

17. **The private modifier can be applied to ...**
    A. A variable
    B. A method
    C. A top level class
    D. All of the above

18. **The abstract modifier can NOT be applied to ...**
    A. A variable
    B. A method
    C. A class
    D. A constructor

19. **A class variable is declared using following**

**modifiers**

A. protected
B. private
C. public
D. static

20. **An unary operator operates on a single value**
    A. True
    B. False

21. **The following types of loop are guaranteed to be executed at least once**
    A. The do loop
    B. The while loop
    C. The for loop
    D. All of the above

22. **The switch() construct is used to make a choice based upon ...**
    A. A char value
    B. An int value
    C. A String value
    D. None of the above

23. **The circumstances that can prevent execution of the code in a finally block are**
    A. The death of the thread
    B. The use of System.exit()
    C. It is always guaranteed to be executed.

24. **Select correct statement(s)**
    A. The continue statement abandons the loop altogether
    B. The break statement causes the current iteration of the loop to be abandoned.
    C. The break statement abandons the loop altogether

25. **How can you declare a overloaded method?**
    A. Reusing the method name with different arguments and same return type
    B. Reusing the method name with different arguments and different

return type
C. Reusing the name with identical arguments and return type
D. None of the above

26. **How can you declare a overriding method?**
    A. Using the same method name with identical arguments and return type
    B. Using the same method name with identical arguments and different return type
    C. Using the same method name with different arguments and same return type
    D. All of the above

27. **When a default constructor is provided by the compiler?**
    A. If you define no constructor at all in a class
    B. When you define at least one constructor
    C. It is always provided whether you define a constructor or not
    D. It is never provided
28. **A static inner class can access ...**
    A. instance variables of the enclosing class
    B. static variables of the enclosing class
    C. Both A and B
    D. None of the above
29. **An inner class created inside a method can access**
    A. Any local variables of a method that contain an inner class.
    B. Any instance variables of the enclosing class
    C. Any final variables of the enclosing class or a method that contain an inner class.
    D. None of the above

30. **Please select true statement(s) for an anonymous inner class**
    A. An anonymous class can not extend to a superclass

B. An anonymous class can not implement an interface
C. An anonymous class can extend to a supperclass and implement an interface both.
D. An anonymous class can extend to a supperclass or implement an interface

31. **Please select true statement(s) for a member inner class**
    A. An inner class in class scope can have any accessibility of the top level class, including private.
    B. An Inner class defined local to a block may be static
    C. An anonymous inner class can not declare a constructor.
    D. An inner class can not have same name as enclosing class.
    E. All of the above

32. **Please select invalid statement(s) for a thread**
    A. You can restart a dead thread
    B. You can't call the methods of a dead thread
    C. Both of the above
    D. None of the above

33. **Select correct statements for a java.lang.String**
    A. Strings are sequences of 16 bit Unicode characters.
    B. Strings are immutable.
    C. Both of the above
    D. None of the above

34. **Select correct statements for == operator.**
    A. It compare object reference of two objects
    B. It compare the value of two objects
    C. Both of the above
    D. None of the above

35. **Please select collection(s) that do not reject duplicates**

A. java.util.List
B. java.util.Set
C. java.util.Map
D. All of the above

36. **Please select a default layout manager of the java.awt.Panel**
    A. java.awt.FlowLayout with left justified
    B. java.awt.FlowLayout with center justified
    C. java.awt.FlowLayout with right justified
    D. None of the above
37. **Please select a default layout manager for the java.awt.Frame**
    A. java.awt.FlowLayout
    B. java.awt.BorderLayout
    C. java.awt.GridBagLayout
    D. None of the above
38. **Please select true statement for prefix operator(++x/--x).**
    A. The prefix operator(++x/--x) evaluates the value of the operand after increment/decrement operation.
    B. The prefix operator(++x/--x) evaluates the value of the operand before increment/decrement operation.
    C. Both A and B
    D. None of the above
39. **Please select true statement(s) for shift operators.**
    A. >>> operator always produces a positive number.
    B. >> always produces a positive number for a positive number.
    C. >> always produces a negative number for a negative number.
    D. None of the above
40. **Please select true statement(s) for shift operators.**
    A. Shifting is not allowed in long integral type
    B. Shifting is not allowed in float integral type
    C. Shifting is not allowed in double integral type
    D. Shifting is not allowed in int integral

type.

41. **Please identify correct assignment for boolean type.**
    A. boolean javaExam=true;
    B. boolean javaExam=True;
    C. boolean javaExam=1;
    D. All of the above
42. **Bitwise operator can be applied to ...**
    A. Integral types
    B. Float types
    C. Both of the above
    D. None of the above
43. **instanceof operator can be used with ...**
    A. interfaces
    B. Arrays
    C. Classes
    D. All of the above
44. **Please select true statement(s)**
    A. The equals method compares content of two objects
    B. The == operator tests reference of two objects
    C. The equals method compares reference of two objects.
    D. The == operator compares content of two objects
45. **Please identify invalid use of comparison operator for the following code:**

```
String s=new String("S");
String t=new String("T");
int x=83;
int y=84;
```

    A. s == t
    B. x!=y
    C. x==s
    D. s!=t
46. **Please select true statement(s) for instanceof operator:**
    A. The instanceof operator can be applied to object reference
    B. The instanceof operator can be applied to an array.
    C. Both of the above
    D. None of the above
47. **please select true statement(s) for static modifier.**

A. A static method can access non-static variables of the class.
B. A static method can call non-static methods.
C. A static method can be overridden by non-static method.
D. None of the above

48. **Please, select true statement(s) for thread.**
A. Invoking a constructor of a Thread registers the thread with thread scheduler.
B. Calling a start() method of thread registers the thread with thread scheduler.
C. Calling a run() method of thread registers the thread with thread scheduler.
D. All of the above.

49. **Invoking yield() on a running thread cause following effect(s):**
A. A running thread will be placed in suspended state.
B. A running thread will be placed in sleeping state.
C. A running thread will be placed in ready state.
D. Neither of the above.

50. **The default priority of the thread is**
A. 0
B. 1
C. 5
D. 10

51. **Which of the following methods are NOT static of Thread class?**
A. start()
B. sleep(long l)
C. run()
D. yield()

52. **Select true statement(s) about an inner class declared inside a method, also known as local inner class**
A. The local inner class can access any local variables declared in the method and the parameters passed to the method.
B. The local inner class can access only

public variables declared in enclosing class

C. The local inner class can access public, private, and protected variables declared in enclosing class.
D. The local inner class can access only final declared variables in the method.

53. **How can you prevent class JavaExam from being extended?**

declare class static JavaExam

declare class synchronized JavaExam

declare class final JavaExam

None of the above

54. **Assume that following methods are declared in one class.**
    1. public void addElement(Object javaExam)
    2. public void addElement( Object [] javaExam)
    3. public Object addElement ( int index, Object javaExam)

    **Please select true statement(s) for above methods.**

. All methods are example of overloading method

All methods are example of overriding method

Method # 1 and method # 2 are example of overloading method, whereas method # 3 is an example of overriding method.

None of the above

55. **When can't you override a method?**

. when method is declared abstract

When method is declared final

when method is declared private

when method is declared static

56. **Please select invalid declaration of the top level class**

. public abstract final class JavaExam

public final class JavaExam implement Runnable

protected static class JavaExam

public class JavaExam extend Thread

57. **Please select invalid types for a switch construct**

. float

long

String

All of the above

58. **Please select invalid java keywords**

. include

ifdef

sizeof

goto

59. **Which of the following statements are NOT true about Java garbage collection Thread?**

. The garbage collection thread has a low priority

The garbage collection thread can be invoked at any time

A call to System.gc() is guaranteed to invoke garbage collection thread immediately

A call to Runtime.getRuntime().gc() is guaranteed to invoke garbage collection thread immediately

60. **An inner class can not declare_____ variables.**

. static

protected

final

transient

61. **Which of the following types can be added to java.util.Vector?**

. reference

null

int

All of the above

62. **Please select a true statement about delete() method of java.io.File.**

. It can delete a file

It can delete an empty directory

Both of the above

Neither of the above

63. **The *continue* statement causes the current iteration of the loop to be skipped and the**

**next iteration starts.**

. True

False

64. **The return type of constructor is void.**

. True

False

65. **'null' is valid java type.**

. True

False

66. **Invoking a constructor of java.io.File class creates a file on the file system.**

True

False

67. **Select true statement(s) about native modifier.**

native modifier can be applied to a class

native modifier can be applied to a method

native modifier can be applied to an instance variable

native variable can be applied to a class variable

68. **What method(s) needed to be declared by a class implementing Runnable interface?**

public void start()

public void run()

public boolean start()

public boolean run()

69. **The priority of a newly created thread is always Thread.NORM_PRIORITY.**

True

False

70. **What methods are declared in *java.lang.Thread* class?**

public static void sleep(long millis, int nanos)

public static native void sleep(long millis, int nanos)

public static native void sleep(long millis)

public static void sleep(long millis)

71. **A *yield* method of the Thread class suspends the operation of the current thread.**

True

False

72. **What methods are NOT synchronized in java.util.Vector class?**

size()

add(int index, Object element)

capacity()

get(int index)

73. **Please select unchecked exception(s)?**

java.lang.NullPointerException

java.lang.ClassNotFoundException

java.lang.ClassCastException

java.awt.AWTException

74. **Which of the following declarations are valid to throw an exception?**

throw new java.lang.Exception();

throws new java.lang.Exception();

Both of the above

None of the above

75. **Which of the following classes are immutable?**

java.lang.String

java.lang.StringBuffer

java.util.Vector

java.lang.Integer

76. **Which of the following classes store and retrieve values based on a key?**

java.util.Hashtable

java.util.Vector

java.util.LinkedList

java.util.HashMap

77. **Which of the following classes can store null value for a key?**

java.util.Hashtable

java.util.HashMap

java.util.Properties

All of the above

78. **java.util.Vector uses synchronized methods to store and retrieve values.**

True

False

79. **java.util.Hashtable uses synchronized methods to store and retrieve values.**

True

False

80. **java.util.HashMap uses synchronized methods to store and retrieve values.**

True

False

81. **Which of the following collections maintain object references in the order they were added?**

java.util.Vector

java.util.Hashtable

java.util.HashMap

java.util.ArrayList

82. **java.util.hashtable implements which of the following interfaces?**

java.util.Dictionary

java.util.Map

java.util.HashMap

java.util.Hashmap

**Answers:**

1. B
2. C
3. C
4. B
5. B
6. B
7. C and D
8. C and F
9. A
10. D. **Comment:** A byte value can represent a range of -128 to +127. Arithmetically answer is 320, but when you store this result to a byte variable you will get a value 64 since result is out of the range (-128 to +127).
11. C
12. A
13. B and C are correct
14. A, B, and C
15. B and C
16. A and B
17. A and B
18. A and D
19. D
20. A. **Comment:** + and - operator can take two values
21. A
22. A and B
23. A and B
24. C
25. A and B
26. A
27. A
28. B
29. C
30. D

31. A , C, and D
32. C.  Thanks Chris Pereira for your feedback in our discussion area.
33. C
34. A
35. A
36. B
37. B
38. A
39. B, and C.
40. B and C
41. A
42. A
43. D
44. A and B
45. C
46. C
47. D
48. B
49. C
50. C
51. A and C
52. C and D
53. C
54. A
55. B
56. A, B, C, and D
57. D
58. A, B, and C
59. B, C, and D
60. A
61. A and B
62. C
63. A
64. B
65. A
66. B
67. B
68. B
69. B **Comment**:  A newly created thread inherits the priority of the Thread that creates it.
70. A and C
71. A
72. A, B, and C
73. A and C
74. A

75. A and D
76. A and D
77. B
78. A
79. A
80. B
81. A and D
82. B

MOCK 7

1. **Consider the following line of code:**
   public class Test{
   public void main(){
   System.out.println("Hi");
   }
   public static void main (String [] args)
   {
   Test t=new Test();
   t.main();
   }
   }
   What will be happen if you compile and run above program?
   A. It will not compile
   B. It will not run
   C. It will compile but will not run
   D. It will output "Hi"

2. **After execution of the code fragment below, what are the value of the variables x1, y1, and z1?**
   int x=10; int y =10; int z=10; int x1, y1, z1;
   x1=++y;
   y1=z++;
   z1=z;
   A. x1 = 10 , y1 = 10 and z1=10
   B. x1 = 10, y1 = 11, and z1=10
   C. x1=10, y1 = 10, and z1=11
   D. x1=11, y1=10, and z1=11

3. **Consider the following application:**
   Class Test{
   public int addTest(int x, int y)

```
{
    x=x+1; y=y+1;
    int z=(x+y);
    return z;
}
public static void main(String [] args)
{
    int x=10; int y=10; int z=0;
    Test t=new Test();
    z= t.addTest(x,y);
    System.out.println("x="+x+", y="+y+", z="+z);
}
}
```

What will be output of the above program?

A. x=10, y=10, z=22

B. x=11, y=11, z=22

C. x=10, y=10, z=20

D. x=11, y=11, z=20

4. **Consider the following application. Assume that MyList class is declared in MyList.java and ListManager class is declared in ListManager.java file.**

```
public class MyList{
int size=1;
public static void main(String [] args)
{
     MyList list=new MyList();
    list.size=10;
    ListManager lm=new ListManager();
    lm.expandList(list);
   System.out.println("list.size="+list.size);
}
}  //end of MyList
public class ListManager{
public void expandList(MyList l)
{
    l.size=l.size+10;
}
}//end of ListManager
```

What will be output of the above program?

. list.size=0

A. list.size=10

B. list.size=11

C. list.size=20

5. **If int x = -1 then which of the following expression results in a positive value in x?**

. x=x>>>32

A. x=x>>>5

B. x=x>>5

C. x=~x

6. **Which of the following lines of code would print "Equal" when you run it?**

. int x=1; float y=1.0F; if(x==y){ System.out.println("Equal");}

A. int x=1; Integer y= new Integer(1); if(x==y) { System.out.println("Equal");}

B. Integer x=new Integer(1); Integer y=new Intger(1); if(x==y){ System.out.println("Equal");}

C. String x="1"; String y="1"; if (x==y) { System.out.println("Equal");}

7. **Which of the following declarations are correct for the top level class?**

. public synchronized class MyTest extends Thread

A. private class MyTest extends Thread

B. public abstract class MyTest extends Thread

C. class MyTest extends Thread

8. **Consider the following lines of code**

```
class Test{
String x;
public void testDemo(int n)
{
    String y;
    if ( n>0) {
        y="Hello";
    }
    System.out.println(x+y);
}
```

```
public static void main(String [] args)
{
        Test test=new Test();
        test.testDemo(2);
}
}
```

What will happen if you try to compile and run above program?

. It will produce compiler warning that variable y may not have been initialized

A. It will produce compiler warning that variable x may not have been initialized

B. It will output "Hello"

C. It will output "nullHello"

9. **Consider that Parent and Child classes are defined in two different files as below:**

```
class Parent{
public Parent(){
System.out.println("I am Parent");
}
}
class Child extends Parent{
public Child(int x){
System.out.println("I am Child");
}
public static void main(String [] args){
Child c=new Child(10);
}
}
```

What will be output if you try to compile and run above program?

. It will not compile.

A. It will compile successfully. It will output "I am Parent" and then "I am Child."

B. It will compile successfully. It will output "I am Child" and then "I am Parent."

C. It will compile successfully, but will not run.

10. **Consider following code:**

```
public class MyList{
static int size;
public expandList(int newSize){
ListExpander lexp=new ListExpander();
Vector expandedList=lexp.expand();
    class ListExpander{
    public Vector expand(){
      Vector v=new Vector(this.size+newSize);
```

```
        return v;
    }
    }
}
}
```

What will happen if you attempt to compile above code?

. compiler error, "cannot refer inside an inner class to a static variable."

A. compiler error, "cannot refer inside an inner class to to a non-final variable newSize defined in a different method."

B. Both of the above

C. None of the above

11. **Consider following code:**

```
public class Parent{
public  int size =0;
static class InnerClass{
public void incrementParentSize(){
   XXX=XXX+10;
}
}
}
```

In above example, how can you access 'size' variable (of outer class Parent) inside innerclass at the place of 'XXX'

. super.size

A. this.size

B. Parent.size

C. Can not access it

12. **Assume that Parent and Child classes are in different files:**

```
public class Parent{
public Parent(int x, int y)
{
     System.out.println("Created Parent");
}
}//end of Parent class
public class Child extends Parent{
public Child(int x, int y){
//
}
public Child(int x, int y, int z){
System.out.println("Creating child");
this(x,y);
```

```
}
public static void main(String [] args){
Child c=new Child(1,2,3);
}
}
```

What will happen if you try to compile and run above program?

. It will compile successfully. It will output "Created Parent" and then "Creating child"

A. It will compile successfully. It will output "Creating child" and then "Created Parent"

B. It will not compile giving warning, "Explicit constructor invocation must be first statement in constructor."

C. It will not compile giving warning, "Expression is not a valid block statement."

13. **Consider following code:**

```
public class OuterClass{
class InnerClass{
}
public void innerClassDemo(){
//Explicit instance of InnerClass
}
}
```

In above example, how can you explicitly create an instance of InnerClass?

. InnerClass i=InnerClass();

A. InnerClass i=OuterClass.InnerClass();

B. InnerClass i=new OuterClass ().new InnerClass();

C. OuterClass.InnerClass i=new OuterClass.InnerClass();

14. **Please select valid array declaration(s):**

. int x[20];

A. int []x=new int[20];

B. int [][] x=new int [20][];

C. int [][][] x=new int[20][20][20];

D. int [] x={1,2,3};

15. **Consider following code:**

```
public class Test{
protected void demo() throws NumberFormatException, ArrayIndexOutOfBoundsException
```

```
{
//something here
}
public void demo(String s){
//something here
}
}//end of Test class
```
Please select true statement(s) for demo method

. It is an example of overloading method

A. It is an example of overriding method

B. Both of the above

C. None of the above

16. **For the following code, please consider that super class is defined in question #15:**

```
public class MyTest extends Test{
private void demo() throws IndexOutOfBoundsException, ClassNotFoundException
{
//something here
}
}//end of MyTest class
```
What will happen if you try to compile above class?

. It will compile successfully.

A. Compiler error: Exception java.lang.ClassNotFoundException in throws clause of void MyTest.demo() is not compatible with void Test.demo().

B. Compiler error: Cannot reduce visibility of the inherited method from Test.

C. Both B and C

17. **Consider the following code:**

```
public class Test{
public void demo(String [] list){
try{
        String s=list[list.length+1];
       System.out.println(s);
   }catch(ArrayIndexOutOfBoundException e){
  return;
 }finally{
     System.out.println("Finally here.");
}
public static void main(String [] args){
Test t=new Test();
String [] list={"one","two"};
t.demo(list);
```

```
System.out.println("Done!");
}
}//end of Test class
```
What happen if you try compile and run above code?

. It will not compile.

A. It will output "null" and then "Finally here."

B. It will output "Done!"

C. It will output "Finally here" and then "Done!"

18. **Please consider following code:**

```
public class Test{
public static void demo(String s)
{
    debug("In demo:"+s);
}
private void debug(String s){
System.out.println(s);
}
public static void main(String [] args){
Test.demo("Hello");
}
}
```
What will happen if you try to compile and run above program?

. It will compile successfully, but will not run.

A. It will compile successfully, and outputs "In demo:Hello."

B. It will not compile with error message "Can not make a static reference to the instance method named."

C. None of the above

19. **Consider the following code:**

```
/** File Drawable.java */
public interface Drawable{
public void draw();
public void  fill();
} /** End of file Drawable.java*/
```

```
/** File Circle.java */
public class Circle implements Drawable{
int center=0;
public void draw(){
System.out.println("Drawing circle");
}
```

```
public static void main(String [] args){
Circle c=new Circle();
c.draw();
}
} /** End of file Circle.java */
```

If you attempt to compile and run Circle class what will be output?

. It will compile successfully, and outputs "Drawing circle."

A. It will not compile, and reports error: "class Circle must implement inherited abstract method void Drawable.fill."

B. It will not compile, and reports error: "Method Drawable.fill requires a body instead of a semicolon."

C. None of the above

20. **Consider the following code:**

```
int x=2; int y=3; int z=4;
if(x>2){
   System.out.println("Tested x");
}if(y<3){
  System.out.println("Tested y");
}if (z<=3){
  System.out.println("Tested z");
}
```

Which line would be part of the output?

. Tested x.

A. Tested y.

B. Tested z.

C. None of the above.

21. **Consider the following code:**

```
for( int i=0;i<2;i++)
{
    for(int j=i;j<3; j++)
   {
         if (i==j)
         {
              continue;
         }
         System.out.println("i="+i+" j="+j);
   }
}
```

Which lines would be part of the output?

. i = 0 j = 1

A. i = 0 j = 2

B. i = 1 j = 2

C. None of the above

22. **Consider the following code:**

```
int j=0;
for( int i=0;i<2;i++)
{
     for (j=i; j<3; j++)
    {
            continue;
    }
 System.out.println("i = "+i+" j = "+j);
}
```

Which lines would be part of the output?

. i = 0 j = 0

A. i = 1 j = 1

B. i = 0 j = 3

C. i = 1 j =3

23. **Consider the following code:**

```
int i=0; int j=0;
for( i=0;i<2;i++)
{
     for (j=i; j<3; j++)
    {
            break;
    }
  System.out.println("i = "+i+" j = "+j);
}
```

Which lines would be part of the output?

. i = 0 j = 0

A. i = 1 j = 1

B. i = 0 j = 3

C. i = 1 j = 3

24. **Consider the following code:**

```
int i, j=0;
```

```
outer:
for( i=0;i<2;i++)
{
     for (j=i; j<3; j++)
    {
            continue outer;
    }
  System.out.println("i = "+i+" j = "+j);
}
```

Which lines would be part of the output?

. i = 0 j = 0

A. i = 1 j = 1

B. i = 0 j = 3

C. None of the above

25. **Consider the following code:**

```
int i, j=0;

for( i=0;i<2;i++)
{
     inner:
     for ( j=i; j<3; j++)
    {
            break inner;
    }
  System.out.println("i = "+i+" j = "+j);
}
```

Which lines would be part of the output?

. i = 0 j = 0

A. i = 1 j = 1

B. i = 0 j = 3

C. None of the above

26. **Consider following lines of code:**

```
Thread currentThread=Thread.currentThread();
int priority = currentThread.getPriority();
Thread t1=new Thread();
t1.setPriority(9);
ThreadGroup tgrp=new ThreadGroup();
tgrp.setMaxPriority(10);
Thread t2=new Thread(tgrp,"t2");
System.out.println("Priority of t1="+t1.getPriority());
```

System.out.println("Priority of t2="+t2.getPriority());
What will be output of the above code?

. Priority of t1=5 and Priority of t2=10

A. Priority of t1=9 and Priority of t2=10

B. Priority of t1=9 and Priority of t2=5

C. Neither of above

27. **Consider the following code:**

```
/** File Thread1.java */
class Thread1 implements Runnable{
public void run(){
System.out.println("Running Thread1");
}}  /** End of file Thread1.java */

/** Thread2.java */
class Thread2 extends Thread{
public void run(){
System.out.println("Running Thread2");
}
public static void main(String [] args){
Thread1 t1= new Thread1();
Thread t2=new Thread2(t1);
t1.start();
t2.start();
}
}
/** End of Thread2.java*/
```

If you try to compile and run above code what will be result?

. "Running thread1" following "Running thread2"

A. "Running thread2" following "Running thread1"

B. It will not compile because in Thread1 and Thread2 start() is not defined .

C. It will not compile because constructor invoked to create Thread2 with arguments (Thread1) is not defined

28. **Consider the following code:**

```
class MyThread extends Thread{
public void run(){
System.out.println("Done");
}
public void demo(){
System.out.println("Demo");
```

```
}
public static void main(String args[]){
MyThread th=new MyThread();
th.run();
th.stop();
th.demo();
}
}
```

What will happen if you try to compile and run above code:

. It will throw an exception at th.run() because run() was called before calling start().

A. It will throw an exception at th.demo() because Thread variable th was already stopped calling stop().

B. It will output "Done" following "Demo"

C. Neither of the above.

29. **Please consider following code:**

```
String s1=" 5 + 5 = 10 ";
s1.trim();
s1.replace('+', '-');
```

How many String objects will be created after executing above lines?

. 1

A. 2

B. 3

C. 4

30.
```
String s="Hi";
StringBuffer sb=new StringBuffer(s);
String s1=new String("There");
StringBuffer sb1=new StringBuffer(s1);
if(s==sb){
System.out.println("s==sb");
}if(s.equals(sb)){
System.out.println("s.equals(sb)");
}if(s1.equals(sb1)){
System.out.println("s1.equals(sb1)");
}
```

Please, select which of the following will be part of output?

. It will not compile at if(s==sb) because operands on both side are not compatible

A. It will print s1.equals(sb1)

B. It will print s.equals(sb)

C. It will compile successfully, but it will not output anything

31. Consider that following code is declared in BussyThread.java file

```
public class BussyThread extends Thread{
    public void run(){
    for(int i=0;i<10; i++){
    i=i-1;
    }//end of for loop
   }//end of run()
public static void main(String args[]){
BussyThread b1=new BussyThread();
BussyThread b2=new BussyThread();
b1.start();
b2.start();
}
}//end of class
```

Above class will start two threads b1 and b2. Select True statements for above class.

. Only b1 thread will get chance to run

A. Only b2 thread will get chance to run

B. Both thread will get chance to run sharing CPU time

C. Neither of the thread will be able to run.

32. **What changes in run() method of BussyThread will enable both threads to run?**

. adding yield() into run method

A. adding try{sleep(1000);}catch (InterruptedException e){} into run method

B. adding wait(1000) into run method

C. Neither of the above

33. **Consider the following classes are in MyThread.java, YourThread.java, and Driver.java files:**

```
public class MyThread implements Runnable{

public void run(){
    System.out.println("Running MyThread");
    }
    }//end of MyThread
    public class YourThread extends Thread{
```

```
public YourThread(Runnable r){
super(r);
}
public void run(){
System.out.println("Running YourThread");
}
}//end of YourThread
public class Driver{

public static void main(String args []){
MyThread t1= new MyThread();
YourThread t2 = new YourThread(t1);
t2.start();
}
}//end of class
```

If you try to run Driver class what will be result?

. It will output "Running MyThread."

A. It will output "Running YourThread."

B. It will output both "Running MyThread," and "Running YourThread."

C. It will not run.

34. Consider following code:

```
35.                          String s=null;
36.                          String t="null";
37.                          if (s==t)
38.                          {
39.                               System.out.println("s equal to
   t");
40.                          }else
41.                          {
42.                          System.out.println("s not equal to t");
43.                          }
```

what will result if you try to compile and run above code?

. it compiles successfully, but throws NullpointerException at if (s==t)

A. It will not compile.

B. It compiles successfully and output "s equal to t"

C. It compiles successfully and output "s not equal to t"

44. Consider the following code:

```
45.               public void demo(){
46.                         String s[];
47.                         if (s.equals(null))
48.                         {
49.                              System.out.println("s is null");
50.                         }else
51.                         {
52.                         System.out.println("s is not equal");
53.                         }
        }
```

What will be result if you try to compile and run above code?

. Compile error produced, "variable s may not have been initialized."

A. It compile successfully, but throws NullpointerException at if ( s.equals(null) )

B. It compile successfully, and outputs "s is null."

C. It compile successfully, and outputs "s is not null."

54. **Consider the following code:**

```
public class MyList
{
private static final int MAX_SIZE = 10;
private Object [] list = new Object[MAX_SIZE];
public void add(Object obj)
{
  int size=list.length;
  if(size >= MAX_SIZE)
  {

     class ListExpander
    {
        public void expand()
       {
            Object temp [] = list;
          list = new Object[size+MAX_SIZE];
          for ( i=0;i<temp.length; i++)
```

```
            {
                list[i]=temp[i];
            }
        }//end of  public void expand()
    } end of class ListExpander
      ListExpander listEx = new ListExpander();
      listExp.expand();
      list[size] = obj;
  }//end of if
}//end of add
}//end of class MyList
```

**What will be result if you try to compile and run the above code:**

. Compiler error reported, "Cannot refer inside an inner class to a non-final variable 'size' defined in a different method."

A. Compiler error reported, "Cannot refer inside an inner class to a private member variable 'list' defined in enclosing class MyList."

B. Compiler error reported, "Cannot refer inside an inner class to a static member variable MAX_SIZE defined in enclosing class MyList."

C. It compiles and runs successfully.

55. **Consider following example of an inner class**

```
public class MyTest{
public String publicVariable = "a";
private String privateVariable = "b";
public static int SIZE = 0;
private static int MAX_SIZE = 0;
public static DemoHelper{
    public demo{
    System.out.println("Demo = "+XXX);
    }
    }
}//end of inner class
}
```

**which variable of the MyTest class will be able to use in place of XXX?**

. publicVariable

A. privateVariable

B. SIZE

C. MAX_SIZE

56. **What will be result if you try to compile and run following code?**
public class Record extends String{}

. Compiler error reported, "Can not extend a final class."

A. Compiler error reported, "Must implement method int compareTo(Object)."

B. Compile and run successfully.

C. None of the above.

57. **Consider the following two classes:**
public class Parent{
protected void demo() throws Exception{}
} // end of Parent class
public class Child extends Parent{
private void demo() {}
}
**What will be result if you try to compile above two classes?**

. Compiler object for the method of a Child class, "Can not reduce the visibility of the inherited method."

A. Compiler object for demo() method of a Child class, "Inherited method is not compatible with void Parent.demo() throws Exception."

B. Compile successfully.

C. None of the above

58. **Consider the following two classes:**
public class Parent{
protected void demo() {}
} // end of Parent class
public class Child extends Parent{
public void demo() throws Exception{}
}
**What will be result if you try to compile above two classes?**

. Compiler object for the method of a Child class, "Can not widen the visibility of the inherited method."

A. Compiler object for demo() method of a Child class, "Exception java.lang.Exception in throws clause of void Child.demo() is not compatible with void Parent.demo()."

B. Compile successfully

C. None of the above

59. **Consider the following two classes:**

```
public class Parent{
protected void demo() {}
} // end of Parent class
public class Child extends Parent{
public int demo()
{return 0;}
}
```

**What will be result if you try to compile above two classes?**

. Compiler object for the method of a Child class, "Can not widen the visibility of the inherited method."

A. Compiler object for the method of a Child class, "Return type is not compatible with void Parent.demo()."

B. Compile successfully.

C. None of the above

60. **Consider the following two classes:**

```
public class Parent{
protected static void demo() {}
} // end of Parent class
public class Child extends Parent{
public void demo() {}
}
```

**What will be result if you try to compile above two classes?**

. Compiler object for the method of a Child class, "Can not widen the visibility of the inherited method."

A. Compiler object for the method of a Child class, "inherited method void Child.demo() is not compatible with void Parent.demo()."

B. Compiler object for the method of a Child class, "The instance method can not override the static method from Parent."

C. Compile successfully.

61. **Consider that class Employee and Salesman are in different file called Employee.java and Salesman.java:**

```
/** Employee.java file*/
public class Employee{
int salary=1000;
public int getSalary(){
return salary;
}
}
/**  End of Employee.java file*/
/** Salesman.java file*/
public class Salesman extends Employee{
int commission =100;
public int getSalary(){
return salary+commission;
}
public static void main(String [] args){
Salesman sm = new Salesman();
Employee em = sm;
System.out.println(em.getSalary());
}
}
/**  End of Salesman.java file*/
```

**What will be result if you try to compile and run above code?**

. Compiler error reported , "Type mismatch: Cannot convert from Salesman to Employee."

A. It compile successfully and outputs 1000.

B. It compiles successfully and outputs 1100.

C. None of the above

62. **Considering following code what will be the result if you try to compile the following code:**

```
public abstract class Test{
public void demo(){
System.out.println("demo");
}
}
```

. It will compile successfully.

A. Compiler error reported, "An abstract method must be defined."

B. Compiler error reported, "Invalid declaration of class."

C. None of the above

63. **Considering following code what will be the result if you try to compile the following code:**

```
public class Test{
public abstract void demo();
}
```

. Compiler error reported, "Method requires a body instead of semicolon."

A. Compiler error reported, "Abstract methods are only defined by abstract classes."

B. Compile successfully.

C. None of the above.

64. **The GenericList has the following method:**

```
public void addItem(Object item)
```

You are writing a class GroceryList that extends GenericList. Which of the following would be legal declarations of overloading methods?

. public void addItem(Vector item)

A. public void addItem(Object [] items) throws Exception

B. protected void addItem(Object item)

C. All of the above

65. **What will be result if you try to compile the following code?**

```
public class Parent{
String name=null;
public Parent(String n){
name=n;
}
}
public class Child extends Parent{
String type="X";
}
```

. Compile successfully.

A. Compiler error reported, because Parent class did not declare constructor with arguments ().

B. Compiler error reported, because Child class did not declare a constructor.

C. Both of the above B and C

66. **What will be legal statement in the following method?**

```
public void demo(int x){
XXX y=10;
}
```

. public int

A. int

B. final int

C. static int

67. **What will be result if you try to compile and run following code fragement?**

```
public void demo (String [] args){
int i=1;
for(int i=0;i<args.length;i++)
{
      System.out.println(args[i]);
}
}
```

. Compile successfully, but throws IndexOutOfBoundException during runtime.

A. Compile error reported, "Local name i is already defined."

B. Throws NullPointerException during runtime

C. None of the above

**Answers:**

4. D

5. D

6. A

7. D

8. B and D

9. A and D

10. C and D

11. A

12. B

13. B

14. D

15. C and D

16. C and D (Corrected)

17. B, C, D and E

18. A

19. D Note: This is an example of overriding method.

20. D

21. C

22. B

23. D
24. A, B, and C
25. C and D
26. A and B
27. D
28. A and B
29. C
30. D
31. C
32. C
33. A
34. C
35. A and B
36. B
37. D
38. A
39. A
40. C and D
41. A
42. A
43. B
44. B
45. C

46. C

47. A

48. B

49. A and B

50. B

51. B and C

52. B

``````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````

1.What is the result when you compile and run the following code?

```
class Top {

   static void myTop() {
      System.out.println("Testing myTop method in Top class");
   }
}
public class Down extends Top {

    void myTop() {
         System.out.println("Testing myTop method in Down class");
    }
    public static void main(String [] args) {
        Top t = new Down();
         t.myTop();
    }
}
```

A) Compile Time error
B) Runtime error
C) Prints Testing myTop method in Top class on the console
D) Prints Testing myTop method in Down class on the screen

2. Which of the code fragments will throw an "ArrayOutOfBoundsException" ?

```
A) for (int i = 0; i < args.length; i ++ ) {
          System.out.print( i ) ;
     }
```
```

B) System.out.print(args.length);
C) for (int i = 0; i < 1; i++) {
        System.out.print(args[i]);
    }
D) None of the above

3. What is the result of the following program, when you compile and run?

```
public class MyTest {

    final int x;
    public MyTest() {
       System.out.println( x + 10 );
    }
    public static void main( String args[] ) {
         MyTest mt = new MyTest();
    }
}
```

A) Compile time error
B) Runtime error
C) Prints on the screen 10
D) Throws an exception

4. What is the output when you compile and run the following code fragment?

```
class MyTest {

  public void myTest() {
       System.out.println("Printing myTest in MyTest class");
  }

  public static void myStat() {
       System.out.println("Printing myStat in MyTest class");
  }
}

public class Test extends MyTest {

   public void myTest() {
       System.out.println("Printing myTest in Test class");
  }

  public static void myStat() {
       System.out.println("Printing myStat in Test class");
  }
```

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

      MyTest mt = new Test();
      mt.myTest();
      mt.myStat();
   }
}
```

A) Printing myTest in MyTest class followed by Printing myStat in MyTest class
B) Printing myTest in Test class followed by Printing myStat in MyTest class
C) Printing myTest in MyTest class followed by Printing myStat in MyTest class
D) Printing myStat in MyTest class followed by Printing myStat in MyTest class

5. Select all the exceptions thrown by wait() method of an Object class, which you can replace in the place of xxx legally?

```
class T implements Runnable {

      public void run() {
            System.out.println( "Executing run() method" );
            myTest();
      }

     public synchronized void myTest() {
            try {
               wait(-1000);
               System.out.println( "Executing the myTest() method" ) ;
            }  XXX
      }
}

public class MyTest {

      public static void main ( String args[] ) {
            T t = new T();
            Thread th = new Thread ( t );
             th.start();
      }
}
```

A) catch (InterruptedException ie) {}
B) catch (IllegalArgumentException il) {}
C) catch (IllegalMonitorStateException im) {}
D) Only catch (InterruptedException e) {} exception

6. Which of the following are examples of immutable classes , select all correct answer(s)?

A) String
B) StringBuffer
C) Double
D) Integer

7. Select the correct answer for the code fragment given below?

```
public class TestBuffer {

    public void myBuf( StringBuffer s, StringBuffer s1) {
          s.append(" how are you") ;
          s = s1;
    }

    public static void main ( String args[] ) {
           TestBuffer tb = new TestBuffer();
           StringBuffer s = new StringBuffer("Hello");
           StringBuffer s1 = new StringBuffer("doing");
           tb.myBuf(s, s1);
           System.out.print(s);
     }
}
```

A) Prints Hello how are you
B) Prints Hello
C) Prints Hello how are you doing
D) Compile time error

8. What is the result when you compile and run the following code?

```
public class MyTest {

  public void myTest( int[] increment ) {
        increment[1]++;
  }

  public static void main ( String args[] ) {
       int myArray[] = new int[1];
       MyTest mt = new MyTest();
       mt.myTest(myArray);
       System.out.println(myArray[1]);
  }
}
```

A) Compile time error
B) Runtime error
C) ArrayOutOfBoundsException
D) Prints 1 on the screen

9. Chose all valid identifiers?

A) int100
B) byte
C) aString
D) a-Big-Integer
E) Boolean
F) strictfp

10. Select the equivalent answer for the code given below?

```
boolean b = true;
if ( b ) {
    x =  y;
} else {
   x = z;
}
```

A) x = b ? x = y : x = z ;
B) x = b ? y : z ;
C) b = x ? y : z ;
D) b = x ? x = y : x = z ;

11. Chose all correct answers?

A) int a [][] = new int [20][20];
B) int [] a [] = new int [20][];
C) int [][] a = new int [10][];
D) int [][] a = new int [][10];

12. Consider the following code and select the correct answer?

```
class Vehicle {

    String str ;
    public Vehicle() {
    }
    public Vehicle ( String s ) {
    str = s;
    }
}

public class Car extends Vehicle {
```

```
     public static void main (String args[] ) {
          final Vehicle v = new Vehicle ( " Hello" );
          v =  new Vehicle ( " How are you");
          v.str = "How is going";
    System.out.println( "Greeting is : "  + v.str );
    }
}
```

A) Compiler error while subclassing the Vehicle
B) Compiler error , you cannot assign a value to final variable
B) Prints Hello
C) Prints How is going

13. Java source files are concerned which of the following are true ?

A) Java source files can have more than one package statements.
B) Contains any number of non-public classes and only one public class
C) Contains any number of non-public classes and any number of public classes
D) import statements can appear anywhere in the class
E) Package statements should appear only in the first line or before any import statements of source file

14. Select all correct answers from the following?

```
int a = -1;
int b = -1;
a  = a >>> 31;
b = b >> 31;
```

A) a = 1, b =1
B) a = -1, b -1
C) a = 1, b = 0
D) a = 1, b = -1

15. What is the value of a , when you compile and run the following code?

```
public class MyTest {

        public static void main ( String args[] ) {

int a = 10;
int b = 9;
int c = 7;
a = a ^ b ^ c;
System.out.println ( a );
```

```
}

}
```

A) 10
B) 9
C) 7
D) 4

16. The following code has some errors, select all the correct answers from the following?

```
public class MyTest {

        public void myTest( int i ) {
               for ( int x = 0; x < i; x++ ) {
                     System.out.println( x ) ;
               }
         }
         public abstract void Test() {
                myTest(10);
         }
}
```

A) At class declaration
B) myTest() method declaration
C) Test() method declaration
D) No errors, compiles successfully

17. At what point the following code shows compile time error?

```
class A {
        A() {
            System.out.println("Class A constructor");
         }
}

class B extends A {
        B() {
            System.out.println("Class B constructor");
         }
}

public class C extends A {
        C() {
            System.out.println("Class C constructor");
         }
```

```
        public static void main ( String args[] ) {
              A a = new A(); // Line 1
              A a1 = new B(); // Line 2
              A a2 = new C(); // Line 3
              B b = new C(); // Line 4
         }
}
```

A) A a = new A(); // Line 1
B) A a1 = new B(); // Line 2
C) A a2 = new C(); // Line 3
D) B b = new C(); // Line 4

18. Which of the following statements would return false? if given the following statements.

String s = new String ("New year");
String s1 = new String("new Year");

A) s == s1
B) s.equals(s1);
C) s = s1;
D) None of the above

19. Select all correct answers about what is the definition of an interface?

A) It is a blue print
B) A new data type
C) Nothing but a class definition
D) To provide multiple inheritance

20. Select all correct answers from the following code snippets?

A)
```
// Comments
import java.awt.*;
package com;
```

B)
```
import java.awt.*;
// Comments
package com;
```

C)
```
package com;
import java.awt.*;
// Comments
```

D)
```
// Comments
package com;
```

```
import java.awt.*;
public class MyTest {}
```

21. What is the result when you compile and run the following code?

```
public class MyError {

      public static void main ( String args[] ) {
            int x = 0;
            for ( int i = 0; i < 10; i++ ) {
                  x = new Math( i );
                 new System.out.println( x );
            }
       }
}
```

A) Prints 0 to 9 in sequence
B) No output
C) Runtime error
D) Compile time error

22. There are two computers are connected to internet, one computer is trying to open a socket connection to read the home page of another computer, what are the possible exceptions thrown while connection and reading InputStream?.

A) IOException
B) MalformedURLException
C) NetworkException
D) ConnectException

23. What is the result from the following code when you run?

```
import java.io.*;

class A {

       A() throws Exception {
             System.out.println ("Executing class A constructor");
             throw new IOException();
        }
}

public class B extends A {
        B() {
             System.out.println ("Executing class B constructor");
         }
```

```
        public static void main ( String args[] ) {
                try {
                A a = new B();
                } catch ( Exception e) {
                        System.out.println( e.getMessage() );
                }
        }
}
```

A) Executing class A constructor
B) Executing class B constructor
C) Runtime error
D) Compile time error

24. What is the result from the following code when you run?

```
import java.io.*;

class A {

       A()  {
            System.out.println ("Executing class A constructor");
        }
       A(int a) throws Exception {
              System.out.println ("Executing class A constructor");
               throw new IOException();
       }

}

public class B extends A {
        B() {
            System.out.println ("Executing class B constructor");
         }

         public static void main ( String args[] ) {
                try {
                A a = new B();
                } catch ( Exception e) {
                        System.out.println( e.getMessage() );
                }
        }
}
```

A) Executing class A constructor followed by Executing class B constructor
B) No output
C) Runtime error
D) Compile time error

25. What is the result when you compile and run the following code?

```
byte Byte = 10;
byte Double = 12;
byte Integer = Byte * Double;
```

A) 120;
B) Compile time error while declaring variables
C) Compile time error while multiplication
D) None of the above

26. Select all valid methods for Component class?

A) setBounds(), setVisible(), setFont()
B) add(), remove()
C) setEnabled(), setVisible()
D)addComponent()

27. Which subclasses of the Component class will display the MenuBar?

A) Window, Applet
B) Applet, Panel
C) Frame
D) Menu, Dialog

28. Select all correct answers from the following statements?

A) Frame's default layout manager is BorderLayout
B) CheckBox, List are examples of non visual components
C) Applets are used to draw custom drawings
D) Canvas has no default behavior or appearance

29. Select all the methods of java.awt.List?

A) void addItem(String s), int getRows()
B) void addItem(String s, int index), void getRows()
C) int[] getSelectedIndexes(), int getItemCount()
D) int[] getSelectedIndexes(), String[] getSelectedItems()

30. Please select all correct answers?

A) java.awt.TextArea.SCROLLBARS_NONE
B) java.awt.TextArea does not generate Key events
C) java.awt.TextField generates Key events and Action events
D) java.awt.TextArea can be scrolled using the <-- and --> keys.

31. What is the result if you try to compile and run the following code ?

```
public class MyTest {

   public static void myTest() {
         System.out.println( "Printing myTest() method" );
   }
   public void myMethod() {
        System.out.println( "Printing myMethod() method" );

   }
   public static void main(String[] args) {
         myTest();
         myMethod();
   }

}
```

A) Compile time error
B) Compiles successfully
C) Error in main method declaration
D) Prints on the screen Printing myTest() method followed by Printing myMethod() method

32. What line of a given program will throw FileNotFoundException?

```
import java.io.*;

public class MyReader {

        public static void main ( String args[] ) {
                try {
                        FileReader fileReader = new FileReader("MyFile.java");
                        BufferedReader bufferedReader = new
BufferedReader(fileReader);
                        String strString;
                        fileReader.close();

                while ( ( strString = bufferedReader.readLine()) != null ) {
                         System.out.println ( strString );
                }

       } catch ( IOException ie) {
                System.out.println ( ie.getMessage() );
         }
```

```
      }
}
```

A) This program never throws FileNotFoundException
B) The line fileReader.close() throws FileNotFoundException
C) At instantiation of FileReader object.
D) While constructing the BufferedReader object

33. When the following program will throw an IOException?

```
import java.io.*;

class FileWrite {
      public static void main(String args[]) {
            try {
            String strString = "Now is the time to take Sun Certification";
            char buffer[] = new char[strString.length()];
            strString.getChars(0, strString.length(), buffer, 0);
            FileWriter f = new FileWriter("MyFile1.txt");
            FileWriter f1 = f;
            f1.close();
            for (int i=0; i < buffer.length; i += 2) {
                f.write(buffer[i]);
            }
           f.close();

          FileWriter f2 = new FileWriter("MyFile2.txt");
          f2.write(buffer);
          f2.close();
          } catch ( IOException ie ) {
              System.out.println( ie.getMessage());
          }
    }
}
```

A) This program never throws IOException
B) The line f1.close() throws IOException
C) While writing to the stream f.write(buffer[i]) throws an IOExcpetion
D) While constructing the FileWriter object

34. Which line of the program could be throwing an exception, if the program is as listed below. Assume that "MyFile2.txt" is a read only file.

Note: MyFile2.txt is read only file..

```
import java.io.*;
```

```
class FileWrite {
      public static void main(String args[]) {
            try {
            String strString = "Updating the critical data section"
            char buffer[] = new char[strString.length()];
            strString.getChars(0, strString.length(), buffer, 0);
            FileWriter f = new FileWriter("MyFile1.txt");
            FileWriter f1 = f;
            for (int i=0; i < buffer.length; i += 2) {
                f1.write(buffer[i]);
            }
           f1.close();

            FileWriter f2 = new FileWriter("MyFile2.txt");
            f2.write(buffer);
            f2.close();
          } catch ( IOException ie ) {
             System.out.println( ie.getMessage());
          }
     }
}
```

A) This program never throws IOException
B) The line f1.close() throws IOException
C) While writing to the stream f1.write(buffer[i]) throws an IOException
D) While constructing the FileWriter f2 = new FileWriter("MyFile2.txt");

35. Select all the correct answers about File Class?

A) A File class can be used to create files and directories
B) A File class has a method mkdir() to create directories
C) A File class has a method mkdirs() to create directory and its parent directories
D) A File cannot be used to create directories

36. Using File class, you can navigate the different directories and list all the files in the those directories?

A) True
B) False

37. Select all the constructor definitions of "FileOutputStream"?

A) FileOutputStream(FileDescriptor fd)
B) FileOutputStream(String fileName, boolean append)
C) FileOutputStream(RandomAccessFile raFile)
D) FileOutputStream(String dirName, String filename)

38. Select all correct answers for Font class?

A) new Font (Font.BOLD, 18, 16)
B) new Font (Font.SERIF, 24, 18)
C) new Font ("Serif", Font.PLAIN, 24);
D) new Font ("SanSerif", Font.ITALIC, 24);
E) new Font ("SanSerif", Font.BOLD+Font.ITALIC, 24);

39. In an applet programing the requirement is that , what ever the changes you do in the applets graphics context need to be accumulated to the previous drawn information. Select all the correct code snippets?

```
A) public void update ( Graphics g) {
        paint( g) ;
    }

B) public void update ( Graphics g) {
        update( g) ;
    }

C) public void update ( Graphics g) {
        repaint( g) ;
    }

D) public void update ( Graphics g) {
        print( g) ;
    }
```

40. How can you load the image from the same server where you are loading the applet, select the correct answer form the following?

```
A) public void init() {
        Image i = getImage ( getDocumentBase(), "Logo.jpeg");
    }

B) public void init() {
        Image i = getImage ( "Logo.jpeg");
    }

C) public void init() {
        Image i = new Image ( "Logo.jpeg");
    }

D) public void init() {
        Image i = getImage ( new Image( "Logo.jpeg") );
    }
```

41. Which of the following answers can be legally placed in the place of XXX?

```
class Check {
       Check() {  }
}

public class ICheck extends Check {
   public static void main ( String[] args) {
        Object o = new ICheck();
        Check i = new ICheck();
        Check c = new Check();

        if ( o instanceof  XXX) {
            System.out.println("True");
     }
   }

}
```

A) Object, ICheck only
B) Check , ICheck only
C) Object only
D) Object, Check, ICheck

42. There are 20 threads are waiting in the waiting pool with same priority, how can you invoke 15th thread from the waiting pool?.

A) By calling resume() method
B) By calling interrupt() method
C) Calling call() method
D) By calling notify(15) method on the thread instance
E) None of the above

43. Select all the correct answers regarding thread synchronization ?

A) You can synchronize entire method
B) A class can be synchronized
C) Block of code can be synchronized
D) The notify() and notifyAll() methods are called only within a synchronized code

44. The thread run() method has the following code, what is the result when the thread runs?

```
try {
    sleep( 200 );
    System.out.println( "Printing from thread run() method" );
} catch ( IOException ie) { }
```

A) Compile time error
B) Prints on the console Printing from thread run() method
C) At line 2 the thread will be stop running and resumes after 200 milliseconds and prints
   Printing from thread run() method
D) At line 2 the thread will be stop running and resumes exactly 200 milliseconds elapsed

45. What is the result when you compile and run the following code?

```
import java.awt.*;

public class TestBorder extends Frame {
    public static void main(String args[]) {
           Button b = new Button("Hello");
           TestBorder t = new TestBorder();
            t.setSize(150,150);
            t.add(b);
    }
}
```

A) A Frame with one big button named Hello
B) A small button Hello in the center of the frame
C) A small button Hello in the right corner of the frame
D) Frame does not visible

46. Select all correct answers from the following?

A) public abstract void Test() { }
B) public void abstract Test();
C) public abstract void Test();
D) native void doSomthing(int i);

47. Please select all correct statements from the following?

A) toString() method is defined in Object class.
B) toString() method is defined in Class class.
C) wait(), notify(), notifyAll() methods are defined in Object class and used for Thread communication.
D) toString() method provides string representation of an Object state.

48. From the following declarations select all correct variables/method declarations?

```
Button bt = new Button ("Hello");
```

A) public transient int val;
B) public synchronized void Test() ;
C) bt.addActionListener(new ActionListener ());
D) synchronized (this) {
    // Assume that "this" is an arbitrary object instance.
  }

49. Which of the following classes will throw "NumberFormatException"?

A) Double
B) Boolean
C) Integer
D) Byte

50. Fill all the blanks from the following ?

A) Math.abs(3.0) returns 3.0
   Math.abs(-3.4) returns --------

B) Math.ceil(3.4) returns --------
   Math.ceil(-3.4) returns -3.0

C) Math.floor(3.4) returns --------
   Math.ceil(-3.4) returns -4.0

D) Math.round(3.4) returns 3
   Math.round(-3.4) returns -------

51. Select from the following which is legal to put in the place of XXX?

```
public class OuterClass {
    private String s = "I am outer class member variable";
    class InnerClass {
        private String s1 = "I am inner class variable";
        public void innerMethod() {
            System.out.println(s);
            System.out.println(s1);
        }
  }
    public static void outerMethod() {
```

```
        // XXX legal code here
        inner.innerMethod();
    }
}
```

A) OuterClass.InnerClass inner = new OuterClass().new InnerClass();
B) InnerClass inner = new InnerClass();
C) new InnerClass();
D) None of the above

52. If you save and compile the following code, it gives compile time error. How do you correct the compile time error?

```
public class OuterClass {
    final String s = "I am outer class member variable";
    public void Method() {
        String s1 = "I am inner class variable";
        class InnerClass {
            public void innerMethod() {
                int xyz = 20;
                System.out.println(s);
                System.out.println("Integer value is" + xyz);
                System.out.println(s1); // Illegal, compiler error
                }
            }
    }

}
```

A) By making s1 as static variable
B) By making s1 as public variable
C) By making s1 as final variable
D) By making InnerClass as static

53. What is the reason using $ in inner class representation?

A) Because the inner classes are defined inside of any class
B) Due to the reason that inner classes can be defined inside any method
C) This is convention adopted by Sun, to insure that there is no ambiguity between packages and inner classes.
D) Because if use getClass().getName() will gives you the error

54. What is the result when you compile and run the following code?

```
import java.util.*;
public class MyVector {
```

```
      public Vector myVector () {
           Vector v = new Vector();
           return v.addElement( "Adding element to vector");
       }

        public static void main ( String [] args) {
             MyVector mv = new MyVector();
                        System.out.println(mv.myVector());
         }
}
```

A) Prints Adding element to vector
B) Compile time error
C) Runtime error
D) Compiles and runs successfully

55. What is the output on the screen when compile and run the following code?

```
public class TestComp {

          public static void main(String args[]) {
                int x = 1;
                System.out.println("The value of x is "  + (~x >> x)  );
           }
 }
```

A) 1
B) 2
C) -1
D) -2

56. The method getWhen() is defined in which of the following class?

A) AWTEvent
B) EventObject
C) InputEvent
D) MouseEvent

57. Select all correct answers from the following?

A) getSource() method is defined in java.awt.AWTEvent class
B) getSource() method is defined in java.util.EventObject class
C) getID() method is defined in java.awt.AWTEvent class
D) getID() method is defined in java.util.EventObject class

58. Which of the following are correct answers?

A) A listener object is an instance of a class that implements a listener interface.
B) An event source is an object , which can register listener objects and sends notifications whenever event occurs.
C) Event sources fires the events.
D) Event listeners fires events.

59. What are possible ways to implement LinkedList class?

A) As a HashMap
B) As a Queue
C) As a TreeSet
D) As a Stack

60. Please select the correct answer from the following?

```
public class ThrowsDemo {
  static void throwMethod() throws Exception {
    System.out.println("Inside throwMethod.");
    throw new IllegalAccessException("demo");
  }
  public static void main(String args[]) {
    try {
      throwMethod();
    } catch (IllegalAccessException e) {
      System.out.println("Caught " + e);
    }
  }
}
```

A) Compilation error
B) Runtime error
C) Compile successfully, nothing is printed.
D) inside throwMethod. followed by caught:
java.lang.IllegalAccessException: demo

Answers

Answer 1:

A) Compile Time error

Explanation:
Generally static methods can be overridden by static methods only .. Static methods may not be overridden by non static methods..

Answer 2:

C) for (int i = 0; i < 1; i++) {
System.out.print(args[i]);
}

Explanation:
Answer C) will cause an "ArrayOutOfBoundsException" if you do not pass the command line arguments to the Java Program. A) and B) will work without any problem.

Answer 3:

A) Compile time error

Explanation:
The final variables behaves like constants, so the final variables must be initialized before accessing them. They can be initialized where they are declared or in every "constructor" if the class. (even if class has one or more constructors defined).

Answer 4:

B) Printing myTest in Test class followed by Printing myStat in MyTest class

Explanation:
Static methods are determined at compile time but the non static (instance methods) methods are identified at runtime using Run Time Type Identification (RTTI).

Answer 5:

A) catch (InterruptedException ie) {}
B) catch (IllegalArgumentException il) {}
C) catch (IllegalMonitorStateException im) {}

Explanation:
The wait() method of an Object class throws InterruptedException when the thread moving from running state to wait state. If the value of timeout is negative or the value of nanos is not in the range 0-999999 then wait() method throws IllegalArgumentException exception at runtime. If the current thread is not the owner of this object's monitor

then it throws IllegalMonitorStateException exception. Click here for more information from Java Documentation.

Answer 6:

A) String
C) Double
D) Integer

Explanation:
String, Integer, Double are immutable classes, once assign a values it cannot be changed. Please refer the wrapper classes for more information on Integer, and Double.

Answer 7:

A) Prints Hello how are you

Explanation:
Assigning or interchanging the object references does not change the values, but if you change the values through object references , changes the values .

Answer 8:

B) Runtime error
C) ArrayOutOfBoundsException

Explanation:
This piece of code throws an ArrayOutOfBoundsException at runtime . If you modify the code int myArray[] = new int[1]; to int myArray[] = new int[2]; , it prints 1 on the screen. The changes you made on the array subscript seen by the caller.

Answer 9:

A) int 100
C) aString
E) Boolean

Explanation:
The byte, strictfp are Java keywords and cannot be defined as identifiers, the a-Big-Integer has "-" which is not a valid identifier. The identifiers must starts with letters, $, or _ (underscore), subsequent characters may be letters, dollar signs, underscores or digits, any other combination will gives you the compiler error.

Answer 10:

B) x = b ? y : z ;

Explanation
If b is true the value of x is y, else the value is z. This is "ternary" operator provides the way to simple conditions into a single expression. If the b is true, the left of (:) is assigned to x else right of the (:) is assigned to x. Both side of the (:) must have the data type.

Answer 11:

A) int a [][] = new int [20][20];
B) int [] a [] = new int [20][];
C) int [][] a = new int [10][];

Explanation:
Multidimensional arrays in Java are just arrays within arrays. Multidimensional arrays are defined as rows and columns. The outer array must be initialized. If you look at the answers the outer arrays are initialized.

Answer 12:

B) Compiler error , you cannot assign a value to final variable

Explanation:
In Java final variables are treated as constants (comparing to other languages like Visual Basic and etc.) , once it is initialized you cannot change the values of primitive, if final variables are object references then you cannot assign any other references.

Answer 13:

B) Contains any number of non-public classes and only one public class
E) Package statements should appear only in the first line or before any import statements of source file

Explanation:
The source files always contains only one package statement, you cannot define multiple package statements and these statements must be before the import statements. At any point of time Java source files can have any number of non-public class definitions and only one public definition class. If you have any import statements those should be defined before class definition and after the package definition.

Answer 14:

D) a = 1, b = -1

Explanation:
The operator >>> is unsigned right shift, the new bits are set to zero, so the -1 is shifted 31 times and became 1 (because a is defined as integer). The operator >> is signed right shift operator, the new bits take the value of the MSB (Most Significant Bit) . The operator << will behave like same as >>> operator. The sifting operation is applicable to only integer data types.

Answer 15:

D) 4

Explanation:
The operator is bitwise XOR operator. The values of a, b, c are first converted to binary equivalents and calculated using ^ operator and the results are converted back to original format.

Answer 16:

C) Test() method declaration

Explanation:
The abstract methods cannot have body. In any class if one method is defined as abstract the class should be defined as abstract class. So in our example the Test() method must be redefined.

Answer 17:

D) B b = new C(); // Line 4

Explanation:
According to the inheritance rules, a parent class references can appear to the child class objects in the inheritance chain from top to bottom. But in our example class B, and class C are in the same level of hierarchy and also these two classes does not have parent and child relationship which violates the inheritance rules.

Answer 18:

A) s == s1
B) s.equals(s1);

Explanation:
The string objects can be compared for equality using == or the equals() method (even though these two have different meaning). In our example the string objects have same wording but both are different in case. So the string object object comparison is case sensitive.

Answer 19:

A) It is a blue print
B) A new data type
D) To provide multiple inheritance

Explanation:
One of the major fundamental change in Java comparing with C++ is interfaces. In Java the interfaces will provide multiple inheritance functionality. In Java always a class can be derived from only one parent, but in C++ a class can derive from multiple parents.

Answer 20:

C) package com;
import java.awt.*;
// Comments

D) // Comments
package com;
import java.awt.*;
public class MyTest {}

Explanation
In a given Java source file, the package statement should be defined before all the import statement or the first line in the .java file provided if you do not have any comments or JavacDoc definitions. The sequence of definitions are:

// Comments (if any)
Package definition
Multiple imports
Class definition

Answer 21:

D) Compile time error

Explanation:
The code fails at the time Math class instantiation. The java.lang.Math class is final class and the default constructor defined as private. If any class has private constructors , we cannot instantiate them from out the class (except from another constructor).

Answer 22:

A) IOException
B) MalformedURLException

Explanation:
In Java the the URL class will throw "MalformedURLException while construncting the URL, and while reading incoming stream of data they will throw IOException..

Answer 23:

D) Compile time error

Explanation:
In Java the constructors can throw exceptions. If parent class default constructor is throwing an exception, the derived class default constructor should handle the exception thrown by the parent.

Answer 24:

A) Executing class A constructor followed by Executing class B constructor

Explanation:
In Java the constructors can throw exceptions. According to the Java language exceptions, if any piece of code throwing an exception it is callers worry is to handle the exceptions thrown by the piece of code. If parent class default constructor is throwing an exception, the derived class default constructor should handle the exception thrown by the parent. But in our example the non default constructor is throwing an exception if some one calls that constructor they have to handle the exception.

Answer 25:

C) Compile time error while multiplication

Explanation:
This does not compile because according to the arithmetic promotion rules, the * (multiplication) represents binary operator. There are four rules apply for binary operators. If one operand is float,double,long then other operand is converted to float,double,long else the both operands are converted to int data type. So in our example we are trying put integer into byte which is illegal.

Answer 26:

A) setBounds(), setVisible(), setFont()
B) add(), remove()
C) setEnabled(), setVisible()

Explanation:
The component class is the parent class of all AWT components like

Button, List, Label and etc. Using these methods you can set the properties of components. The add(), remove() methods are used to add PopMenu and to remove MenuComponent.

Answer 27:

C) Frame

Explanation:
Java supports two kinds of menus, pull-down and pop-up menus. Pull-down menus are accessed are accessed via a menu bar. Menu bars are only added to Frames.

Answer 28:

A) Frame's default layout manager is BorderLayout
D) Canvas has no default behavior or appearance

Explanation:
In Java AWT each container has it's own default layout manager implemented as part of implementation. For example Frame has default layout manager is BorderLayout , Applet has FlowLayout and etc. The Canvas is kind of component where you can draw custom drawings. The Canvas generates Mouse, MouseMotion, and Key events .

Answer 29:

A) void addItem(String s), int getRows()
C) int[] getSelectedIndexes(), int getItemCount()
D) int[] getSelectedIndexes(), String[] getSelectedItems()

Explanation:
The java.awt.List has methods to select , count the visible rows.
void addItem(String s) --> adds an item to the bottom of the list
int getRows() --> returns the number of visible lines in the list
int[] getSelectedIndexes() --> returns array of indexes currently selected items
int getItemCount() --> returns the number of items in the list
String[] getSelectedItems() --> returns array of string values of currently selected items

Answer 30:

A) java.awt.TextArea.SCROLLBARS_NONE
C) java.awt.TextField generates Key events and Action events
D) java.awt.TextArea can be scrolled using the <-- and --> keys.

Explanation:
The TextArea and TextField are the subclasses of TextComponent class. The TextArea has static fields to give you the functionality of horizontal and vertical scroll bars. These are the following fields:
java.awt.TextArea.SCROLLBARS_BOTH
java.awt.TextArea.SCROLLBARS_NONE
java.awt.TextArea.SCROLLBARS_HORIZONTAL_ONLY
java.awt.TextArea.SCROLLBARS_VERTICAL_ONLY
The TextArea and TextField will generate Key events and TextField will generate Action events apart from the Key events.

Answer 31:

A) Compile time error

Explanation:
In Java there are two types of methods , static and non static methods. Static methods are belong to class and non static methods are belongs to instances. So from a non static method you can call static as well as static methods, but from a static method you cannot call non static methods (unless create a instance of a class) but you can call static methods.

Answer 32:

C) At instantiation of FileReader object.

Explanation:
While constructing the FileReader object, if the file is not found in the file system the "FileNotFoundException" is thrown. If the input stream is closed before reading the stream throws IOException.

Answer 33:

C) While writing to the stream f.write(buffer[i]) throws an IOException

Explanation:
While writing to a IO stream if the stream is closed before writing throws an IOException. In our example the f (stream) is closed via f1 reference variable before writing to it.

Answer 34:

D) While constructing the FileWriter f2 = new FileWriter("MyFile2.txt");

Explanation:
Constructing the FileWriter object, if the file already exists it overrides it (unless explicitly specified to append to the file). FileWriter will create the

file before opening it for output when you create the object. In the case of read-only files, if you try to open and IOException will be thrown.

Answer 35:

A) A File class can be used to create files and directories
B) A File class has a method mkdir() to create directories
C) A File class has a method mkdirs() to create directory and its parent directories.

Explanation:
The File class has three methods to create empty files, those are createNewFile(), createTempFile(String prefix, String suffix) and createTempFile(String prefix, String suffix, File directory).
File class has two utility methods mkdir() and mkdirs() to create directories. The mkdir() method creates the directory and returns either true or false. Returning false indicates that either directory already exists or directory cannot be created because the entire path does not exists. In the situation when the path does not exists use the mkdirs() method to create directory as well as parent directories as necessary.

Answer 36:

A) True

Explanation:
File class can be used to navigate the directories in the underlying file system. But in the File class there is no way you change the directory . Constructing the File class instance will always point to only one particular directory. To go to another directory you may have to create another instance of a File class.

Answer 37:

A) FileOutputStream(FileDescriptor fd)
B) FileOutputStream(String fileName, boolean append)

Explanation:
The valid FileOutputStream constructors are:

- FileOutputStream(String fileName)
- FileOutputStream(File file)
- FileOutputStream(FileDescriptor fd)
- FileOutputStream(String fileName, boolean append)

Answer 38:

C) new Font ("Serif", Font.PLAIN, 24);
D) new Font ("SanSerif", Font.ITALIC, 24);
E) new Font ("SanSerif", Font.BOLD+Font.ITALIC, 24);

Explanation:
The Font class gives you to set the font of a graphics context. While constructing the Font object you pass font name, style, and size of the font. The font availability is dependent on platform. The Font class has three types of font names called " Serif", "SanSerif", Monospaced" these are called in JDK 1.1 and after "Times Roman", Helavatica" and "Courier".

Answer 39:

```
A) public void update ( Graphics g) {
        paint( g) ;
    }
```

Explanation:
If you want accumulate the previous information on the graphics context override the update() and inside the method call the paint() method by passing the graphics object as an argument. The repaint() method always calls update() method

Answer 40:

```
A) public void init() {
          Image i = getImage ( getDocumentBase(), "Logo.jpeg");
    }
```

Explanation:
The Applet and Toolkit classes has a method getImage() , which has two forms:

- getImage(URL file)
- getImage(URL dir, String file)

These are two ways to refer an image in the server . The Applet class getDocumentBase() methods returns the URL object which is your url to the server where you came from or where your image resides.

Answer 41:

D) Object, Check, ICheck

Explanation:
The instanceof operator checks the class of an object at runtime. In our example o refers to Object class and Check and ICheck refers to the

subclasses of Object class. Due to the inheritance hierarchy Check and ICheck returns true.

Answer 42:

E) None of the above

Explanation:
There is no way to call a particular thread from a waiting pool. The methods notify() will calls thread from waiting pool, but there is no guaranty which thread is invoked. The method notifyAll() method puts all the waiting threads from the waiting pool in ready state.

Answer 43:

A) You can synchronize entire method
C) Block of code can be synchronized
D) The notify() and notifyAll() methods are called only within a synchronized code

Explanation:
The keyword controls accessing the single resource from multiple threads at the same time. A method or a piece of code can be synchronized, but there is no way to synchronize a calls. To synchronize a method use synchronized keyword before method definition. To synchronize block of code use the synchronized keyword and the arbitrary instance.

Answer 44:

A) Compile time error

Explanation:
The IOException never thrown here. The exception is thrown is InterruptedException. To correct instead of catching IOException use InterruptedException.

Answer 45:

D) Frame does not visible

Explanation:
The Frame is not going to be visible unless you call setVisible(true) method on the Frame's instance. But the frame instance is available in computers memory. If do not set the size of the Frame you see default size of the frame (i.e.. in minimized mode)

Answer 46:

C) public abstract void Test();
D) native void doSomthing(int i);

Explanation:
The abstract methods does not have method bodies. In any given class if one method is defined as abstract the class must defined as abstract class.

Answer 47:

A) toString() method is defined in Object class.
C) wait(), notify(), notifyAll() methods are defined in Object class and used for Thread communication.
D) toString() method provides string representation of an Object state.

Explanation:
The toString() is defined in Object class the parent all classes which will gives you the string representation of the object's state. This more useful for debugging purpose. The wait(), notify(), notifyAll() methods are also defined in Object class are very helpful for Thread communication. These methods are called only in synchronized methods.

Answer 48:

A) public transient int val;
D) synchronized (this) {
    // Assume that "this" is an arbitrary object instance.
  }

Explanation:
To define transient variables just include "transient" keyword in the variable definition. The transient variables are not written out any where, this is the way when you do object serialization not to write the critical data to a disk or to a database.

The "synchronized" keyword controls the single resource not to access by multiple threads at the same time. The synchronized keyword can be applied to a method or to a block of code by passing the arbitrary object instance name as an argument.

Answer 49:

A) Double
C) Integer
D) Byte

Explanation:
In Java all the primitive data types has wrapper classes to represent in

object format and will throw "NumberFormatException". The Boolean does not throw "NumberFormatException" because while constructing the wrapper class for Boolean which accepts string also as an argument.

Answer 50:

A) Math.abs(3.0) returns 3.0
Math.abs(-3.4) returns 3.4

B) Math.ceil(3.4) returns 4.0
Math.ceil(-3.4) returns -3.0

C) Math.floor(3.4) returns 3.0
Math.floor(-3.4) returns -4.0

D) Math.round(3.4) returns 3
Math.round(-3.4) returns -3

Explanation:
The Math class abs() method returns the absolute values, for negative values it just trips off the negation and returns positive absolute value. This method returns always double value.

The method ceil(), returns double value not less than the integer (in our case 3). The other ways to say this method returns max integer value . (All the decimals are rounded to 1 and is added to integer value). For negative values it behaves exactly opposite.

The method floor() is exactly reverse process of what ceil() method does.

The round() method just rounds to closest integer value.

Answer 51:

A) OuterClass.InnerClass inner = new OuterClass().new InnerClass();

Explanation:
The static methods are class level methods to execute those you do not need a class instance. If you try to execute any non static method or variables from static methods you need to have instance of a class. In our example we need to have OuterClass reference to execute InnerClass method.

Answer 52:

C) By making s1 as final variable

Explanation:
In Java it is possible to declare a class inside a method. If you do this there are certain rules will be applied to access the variables of enclosing class and enclosing methods. The classes defined inside any method can access only final variables of enclosing class.

Answer 53:

C) This is convention adopted by Sun , to insure that there is no ambiguity between packages and inner classes.

Explanation:
This is convention adopted to distinguish between packages and inner classes. If you try to use Class.forName() method the call will fail instead use getCLass().getName() on an instance of inner class.

Answer 54:

B) Compile time error

Explanation:
The method in Vector class , addElement() returns type of void which you cannot return in our example. The myVector() method in our MyVector class returns only type of Vector.

Answer 55:

C) -1

Explanation:
Internally the x value first gets inverted (two's compliment) and then shifted 1 times. First when it is inverted it becomes negative value and shifted by one bit.

Answer 56:

C) InputEvent

Explanation:
The InputEvent class has method getWhen() which returns the time when the event took place and the return type is long.

Answer 57:

B) getSource() method is defined in java.util.EventObject class
C) getID() method is defined in java.awt.AWTEvent class

Explanation:
The super class of all event handling is java.util.EventObject which has a method called getSource() , which returns the object that originated the event.

The subclass of EventObject is AWTEvent has a method getID() , which returns the ID of the event which specifies the nature of the event.

Answer 58:

A) A listener object is an instance of a class that implements a listener interface.
B) An event source is an object , which can register listener objects and sends notifications whenever event occurs.
C) Event sources fires the events.

Explanation:
The event listeners are instance of the class that implements listener interface . An event source is an instance of class like Button or TextField that can register listener objects and sends notifications whenever event occurs.

Answer 59:

B) As a Queue
D) As a Stack

Explanation:
This implements java.util.List interface and uses linked list for storage. A linked list allows elements to be added, removed from the collection at any location in the container by ordering the elements. With this implementation you can only access the elements in sequentially.You can easily treat the LinkedList as a stack, queue and etc., by using the LinkedList methods.

Answer 60:

A) Compilation error

Explanation:
The method throwMethod() is throwing and type Exception class instance, while catching the exception you are catching the subclass of Exception class.