

Q1:

```
class TestObj {  
    public static void main(String[] a) {  
        Object o=new Object(){  
            public boolean equals(Object obj){  
                return true;  
            }  
        };  
        System.out.print(o.equals("Test"));  
    }  
}
```

Correct Answer : 3

What happens when you try to compile and run the above code?

1. Compilation error in Line 3
2. Compilation Error in line 4
3. True
4. False

Q2 : Fill the blocks in the code using options given below so that it displays each word in string st on a separate line.

```
String st = "SCJP stands for Sun Certified Java Programmer";
```

```
_____ words = st.split _____ ;  
_____ ( String w : _____ )  
System.out.println( _____ );
```

w

String[]

String

words

words[]

()

for

foreach

Q2 : Fill the blocks in the code using options given below so that it displays each word in string st on a separate line.

```
String st = "SCJP stands for Sun Certified Java Programmer";
```

```
String[] words = st.split (   ) ;  
for ( String w : words )  
System.out.println( w );
```

w

String[]

String

words

words[]

()

for

foreach

Q3:

```
class Plant {  
String getName() { return "plant"; }  
Plant getType() { return this; }  
}
```

```
class Flower extends Plant {  
// insert code here  
}
```

```
class Tulip extends Flower {  
}
```

Correct Answer : 1, 3, 4

Which statement(s), inserted at line 6, will compile? (Choose Three)

1. Flower getType() { return this; }
2. String getType() { return "this"; }
3. Plant getType() { return this; }
4. Flower getType() { return new Tulip(); }

Q4: Which statement(s) are true? (Choose one)

1. Cohesion is the OO principle most closely associated with hiding implementation details.
2. Cohesion is the OO principle most closely associated with making sure that classes know about other classes only through their APIs.
3. Cohesion is the OO principle most closely associated with making sure that a class is designed with a single, well-focused purpose.
4. Cohesion is the OO principle most closely associated with allowing a single object to be seen as having many types.

Correct Answer : 3

Q5:

```
import java.util.*;
class SortList {
    public static void main(String[] args) {
        List<String> x = new ArrayList<String>();
        x.add(" x"); x.add("xx"); x.add("Xx");

        // insert code here
        for(String s: x) System.out.println(s);
    }
}
```

Correct Answer : 5

And the output:

```
xx
Xx
x
```

Which code, inserted at // insert code here, will produce the preceding output? (Choose One.)

1. Collections.sort(x);
2. Comparable c = Collections.reverse(); Collections.sort(x,c);
3. Comparator c = Collections.reverse(); Collections.sort(x,c);
4. Comparable c = Collections.reverseOrder(); Collections.sort(x,c);
5. Comparator c = Collections.reverseOrder(); Collections.sort(x,c);

Q6 :

```
public static void before () {
    Set set = new TreeSet();
    set.add("2");
    set.add(3);
    set.add("1");
    Iterator it = set.iterator();

    while (it.hasNext())
        System.out.print(it.next() + " ");
}
```

Correct Answer : 4

Which of the following statements are true?

1. The before() method will print 2 3 1
2. The before() method will print 1 2 3
3. The before() method will not compile.
4. The before() method will throw an exception at runtime.

Q7:

```
class First {
    public static void classMethod() { System.out.println("classMethod() in First"); }
    public void instanceMethod() { System.out.println("instanceMethod() in First"); }
}
class Second extends First {
    public static void classMethod() {System.out.println("classMethod() in Second"); }
    public void instanceMethod() { System.out.println("instanceMethod() in Second");}
}

class StaticOverride {
    public static void main(String[] args) {
        First f = new Second();
        f.instanceMethod();
        f.classMethod();
    }
}
```

Correct Answer : 2

What is the output of the above program?

1. instanceMethod() in First, classMethod() in Second()
2. instanceMethod() in Second, classMethod() in Second
3. instanceMethod() in Second, classMethod() in First
4. instanceMethod() in First, classMethod() in First()
5. Compilation error as it is not possible to override static methods
6. Compilation error as you are trying to call static method with an object

Q8:

```
enum Languages {
    JAVA("Sun Microsystems"),
    CSHARP("Microsoft"),
    PLSQL("ORACLE");

    private String company;
    public String getCompany() {
        return company;
    }
    Languages(String company) {
        this.company = company;
    }
}
```

Correct Answer : 4

```
public class EnumQuestions {
    public static void main(String[] args) {
        System.out.println( Languages.JAVA + " Comes From " +
            Languages.JAVA.getCompany());
    }
}
```

What is the output of the above program?

1. Compilation error as you cannot declare variable Enum
2. Compilation error as you cannot have methods in enum
3. Compiles but throws exception at runtime.
4. Compiles and displays - Java Comes From Sun Microsystems

Q9 :

Given two files :

```
1. class One {
2.   public static void main(String[] args) {
3.     int assert = 0;
4.   }
5. }
```

```
1. class Two {
2.   public static void main(String[] args) {
3.     assert(false);
4.   }
5. }
```

Correct Answer : 2,6

And the four command-line invocations:

```
javac -source 1.3 One.java
javac -source 1.4 One.java
javac -source 1.3 Two.java
javac -source 1.4 Two.java
```

What is the result? (Choose Two)

1. Only one compilation will succeed.
2. Exactly two compilations will succeed.
3. Exactly three compilations will succeed.
4. All four compilations will succeed.
5. No compiler warnings will be produced.
6. At least one compiler warning will be produced.

Q10:

Given:

```
10. public static void main(String[] args) {
11.     Queue<String> q = new LinkedList<String>();
12.     q.add("Veronica");
13.     q.add("Wallace");
14.     q.add("Duncan");
15.     showAll(q);
16. }
17.
18. public static void showAll(Queue q) {
19.     q.add(new Integer(42));
20.     while (!q.isEmpty ( ))
21.         System.out.print(q.remove( ) + " ");
22. }
```

Correct Answer : 2

What is the result?

1. Veronica Wallace Duncan
2. Veronica Wallace Duncan 42
3. Duncan Wallace Veronica
4. 42 Duncan Wallace Veronica
5. Compilation fails.
6. An exception occurs at runtime.

Q11:

class Numbers implements Runnable

```
{
  Thread t; // line 2
  public static void main(String args[]) {
    t = new Thread(); // line 4
    t.start(); // line 5
  }
  public void run() {
    for( int i = 1 ; i <= 10 ; i ++ )
      System.out.printf("%s : %d\n", Thread.currentThread().getName(), i);
  }
}
```

Correct Answer : 2, 3

Which of the following changes must be applied to above program to make it compile and run successfully (Choose two).

1. Change line 2 to Runnable t;
2. Change line 2 to static Thread t;
3. Change line 4 to t = new Thread(new Numbers());
4. Change line 4 to t = new Thread(this)
5. Change line 5 to t.run()

Q12:

```
class TestEx {
public static void main(String args[]) {
try {
int n;
String s = args[0];
n = 100 / Integer.parseInt(s); // line 6
}
finally {
System.out.println(n); // line 9
}
// line 11
}
}
```

Correct Answer : 2

When you try to compile and then run the above program with the following inputs, what happens.

```
java TestEx 10
java TestEx 0
```

1. Compilation error at line 6
2. Compilation error at line 9
3. In both cases control does NOT reaches line 11
4. In both cases control reaches line 11
5. Outputs 10 and then displays error

Q 13:

```
public class TwoThreads {
static Thread first, second;
public static void main(String[] args) {
first = new Thread() {
public void run() {
System.out.println("A");
try { second.sleep(1000); }
catch (Exception e) { System.out.println("B"); }
System.out.println("C");
}
};
second = new Thread() {
public void run() {
System.out.println("D");
try { first.wait(); }
catch (Exception e) { System.out.println("E"); }
System.out.println("F");
}
};
first.start();
second.start();
}
}
```

Correct Answer : 2, 3, 4

Which letters will eventually appear somewhere in the output? (Choose Three)

1. B
2. C
3. E
4. F

Q14:

```
class TestDivision
{
    public static void main(String ... args) {
        int i = 10;
        double d = 20;

        print(d,0);
        print(i,0);
    }
    public static void print(int x,int y)
    {
        System.out.print( x / y);
    }

    public static void print(double x,double y)
    {
        System.out.print( x / y);
    }
}
```

Correct Answer : 3

What is the output of the above program?

1. Runtime Exception
2. Infinity Infinity
3. Infinity followed by Runtime Exception
4. Compilation Error
5. Two Runtime Exceptions

Q 15:

Which are true about a static nested class? (Choose Two)

1. You must have a reference to an instance of the enclosing class in order to instantiate it.
2. It does not have access to non-static members of the enclosing class.
3. Its variables and methods must be static.
4. If the outer class is named MyOuter, and the nested class is named MyInner, it can be instantiated using new MyOuter.MyInner();
5. It must extend the enclosing class.

Correct Answer : 2, 4

Q16:

Which are true about an anonymous inner class? (Choose One)

1. It can extend exactly one class and implement exactly one interface.
2. It can extend exactly one class and can implement multiple interfaces.
3. It can extend exactly one class or implement exactly one interface.
4. It can implement multiple interfaces regardless of whether it also extends a class.
5. It can implement multiple interfaces if it does not extend a class.

Correct Answer : 3

Q17:

```
public class TestThis {  
    public static void main(String[] args) {  
        B b = new B("Test");  
    }  
}
```

```
class A {  
    A() { this("1", "2"); }  
    A(String s, String t) { this(s + t); }  
    A(String s) { System.out.println(s); }  
}
```

```
class B extends A {  
    B(String s) { System.out.println(s); }  
    B(String s, String t) { this(t + s + "3"); }  
    B() { super("4"); }  
}
```

Correct Answer : 4

What is the output of the above program?

1. It will simply print Test.
2. It will print Test followed by Test.
3. It will print 123 followed by Test.
4. It will print 12 followed by Test.
5. It will print 4 followed by Test.

Q18:

```

public class ThreadTest {
    public static void main (String args[]) {
        final String s="s";
        final Thread t= new Thread (new Runnable ()
        {
            public void run () {
                try {
                    for (int i=1; i < 5; i++) {
                        synchronized (this) {
                            System.out.print (" " + i);
                            this.wait ();
                        }
                    }
                } catch (Exception ex) { ex.printStackTrace (); }
            }
        });
        Thread t2 = new Thread (new Runnable ()
        {
            public void run () {
                while (true) {
                    synchronized (this) { this.notify (); }
                }
            }
        });
        t.start (); t2.start ();
    }
}

```

Correct Answer : 5

What is the output of the above program?

1. A compilation error will occur
2. 1 2 3 4 will be printed
3. run-time error will be thrown
4. Nothing will be printed to the output
5. 1 will be printed

Q19:

```

public class BoxingTest {
    public static void main(String[] args) {
        byte b = 10;
        print(b);

        Integer i = 10;
        print(i);
    }
    public static void print(int n) {
        System.out.println("Primitive int");
    }
    public static void print(Byte n) {
        System.out.println("Wrapper class byte");
    }
    public static void print(Object o) {
        System.out.println("Object class");
    }
}

```

Correct Answer : 3

What will be the output of the above program?

1. Primitive int, Primitive int
2. Object class, Object class
3. Primitive int, Object class
4. Wrapper class byte, Object class
5. Wrapper class byte, Primitive Int

Q20:

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

        Integer i1 = 10;
        Integer i2 = 10;

        Integer i3 = 1000;
        Integer i4 = 1000;

        System.out.println( i1 == i2);
        System.out.println( i1.equals(i2));

        System.out.println( i3 == i4);
        System.out.println( i3.equals(i4));

    }
}
```

Correct Answer : 4

What will be the output of the above program?

1. False True False True
2. True True False False
3. False True False False
4. True True False True

Q 21:

```
import java.util.*;
class Product {
    String name;
    int price;
    public Product(String name, int price) {
        this.name = name;
        this.price = price;
    }
    public int compareTo(Object o) {
        Product p = (Product) o;
        return p.name.compareTo(name);
    }
}
```

Correct Answer : 3

```
class TestProduct {
    public static void main(String args[]) {
        ArrayList al = new ArrayList();
        al.add(new Product("p1",100));
        al.add(new Product("p2",200));
        Collections.sort(al);
    }
}
```

Which of the following changes independently would allow al to be sorted.

1. Make Product class implement Comparator interface
2. Change ArrayList to ArrayList<Product>
3. Make Product class implement Comparable interface
4. Change method compareTo(Object o) to compareTo(Product p)

Q22:

```
import java.util.*;
class Employee { }
class Manager extends Employee { }
class Director extends Manager { }

class WildCards {
    public static void main(String args[]) {
        List<Employee> el = new ArrayList<Employee>();
        List<Manager> ml = new ArrayList<Manager>();
        List<Director> dl = new ArrayList<Director>();

        print(el); // line 17
        print(dl); // line 18
        print(ml); // line 19

    }
    public static void print(List<? extends Manager> lst)
    {
    }
}
```

Which of the above lines cause compile time error?

1. only line 17
2. only line 18
3. line 17 and 18
4. line 18 and 19
5. None of the lines
6. All lines

Correct Answer : 4

Q23:

Which of the following are true about serialization? (Choose Three)

1. The entire object graph is serialized
2. Static members are not serialized
3. If super class is serializable then sub classes are also serializable
4. Object class implements Serializable interface
5. When you serialize a collection, only serializable object in the collection are serialized and other are ignored.

Correct Answer : 1,2,3

Q24:

```
public class ImmutableStrings
{
    public static void main(String[] args)
    {
        String one = "someString";
        String two = "someString";

        System.out.println(one.equals(two));
        System.out.println(one == two);
    }
}
```

Correct Answer : 1

What is the output of the above program?

1. True True
2. False False
3. True False
4. False True
5. Compilation Error
6. Runtime Exception

Q25:

Which of the following are NOT methods of BufferedReader class? (Choose two)

1. flush()
2. readLine()
3. reset()
4. equals()
5. unread()

Correct Answer : 1,5