1. Q. What will be the output?

```
public class Test {
   public static void main(String[] args) {
      try {
         badMethod();
         System.out.print("E");
         return;
      } catch (RuntimeException ex) {
         System.out.print("B");
      } catch (Exception ex1) {
         System.out.print("C");
      } finally {
         System.out.print("D");
      }
   }
   public static void badMethod() {
      System.out.print("A");
   }
}
```

1). **AED**

2). ABD

3). AE

4). Compilation Error

Solution :
option [1] is correct

2. Q.

```
class Atom {
        Atom() {
      System.out.print("atom ");
}
 }
//////////////////////
class Rock extends Atom {
Rock(String type) {
System.out.print(type);
}
}
//////////////////////

public class Mountain extends Rock {
Mountain() {
super("granite ");
new Rock("granite ");
}
public static void main(String[] a) {
new Mountain();
}
}
```

What will be the result?
1). Compilation fails.

2). atom granite

3). **atom granite atom granite**

4). atom granite granite

Solution :
option [3] is correct

3.
Q.Which two code fragments, inserted independently at line 3, generate the output 4247? (Choose t wo.)

```
1.public class TestString3 {
2.      public static void main(String[] args) {
3.              // insert code here
4.            System.out.println(s);
5.      }
6. }
```

1). String s = "123456789"; s = (s-"123").replace(1,3,"24") - "89";

2). **StringBuffer s = new StringBuffer("123456789"); s.delete(0,3).replace(1,3,"24").delete(4,6);**

3). StringBuffer s = new StringBuffer("123456789"); s.substring(3,6).delete(1,3).insert(1, "24");

4). StringBuilder s = new StringBuilder("123456789"); s.substring(3,6).delete(1,2).insert(1, "24");

5). **StringBuilder s = new StringBuilder("123456789");**
s.delete(0,3).delete(1,3).delete(2,5).insert(1, "24");

Solution :
option [2,5] are correct

4. Q. What is the output of the following?

```
1  class StringCheck {
2
3    static String[] str=new String[50];
4    public static void main(String[] args) {
5        for(int i=0; i<str.length; i++) {
6           str[i] = i;
7           System.out.print(str[i]);
8        }
9    }
10 }
```

1). **compile error at line 6.**

2). 0 to 49 printed continuously.

3). compile error at line 7.

4). null printed 50 times.

Solution :
option [1] is correct

5. Q. What is the output if main() is run?

```
public abstract class Vehicle {
 private int tyres;
 public void setTyres(int tyres) {
   this.tyres = tyres;
 }
 public int getTyres() {
   return tyres;
 }
}
public class Car extends Vehicle {
  @Override
  public int getTyres() {
     return super.getTyres()+1;
  }
}
public class Main {
  public static void main(String args[]) {
    Car c = new Car();
    c.setTyres(5);
    System.out.println("Tyres = "+c.getTyres());
  }
}
```

1). Tyres = 5

2). **compilation error**

3). Tyres = 6

4). runtime exception

Solution :
option [2] is correct

6. Q. What is the output for the following?

```
   public class Test {
      public static void main(String [] args) {
         short a, b, c=0;
         a=1;
         b=2;
         c=a+b;
         System.out.println(c);
      }
  }
```

1). **Compile error.**

2). **3**

3). 1

4). runtime exception

Solution :
option [1] is correc

7. Q. What is the output?

```
String k ="big ";
k.concat("crowded ");
k += "city";
System.out.println(k);
```

1). compile error

2). big crowded city

3). big crowded

4). **big city**

Solution :
option [4] is correct

8. Q. 1. interface Foo {}
2. class Alpha implements Foo {}
3. class Beta extends Alpha {}
4. class Delta extends Beta {
5. public static void main(String[] args) {
6. Beta x = new Beta();
7. // insert code here Line 7.
8. }
9. }

Which code, inserted at line 7., will cause a java.lang.ClassCastException?

1). Alpha a = x;

2). **Foo f = (Delta)x;**

3). Foo f = (Alpha)x;

4). Beta b = (Beta)(Alpha)x;

Solution :
option [2] is correct
Attempted :

9. Q. public class Test {

```
public static void main(String[] args) {
    int n = 1;
    outer :
    while (n < 6) {
        for (int i = 0; i < 10; i++) {
            if (i % 2 == 0) {
                System.out.print(i + " ");
                continue;
            } else {
                System.out.print(n + " ");
                break outer;
            }
        }
```

```
            n++;
        }
    }
}
```

1). 0 2 4 6 7 1

2). 0 1 0 1 0 1 0 1 0 1

3). **0 1**

4). compile Erorr

Solution :
option [3] is correct

10. Q. What will be the output?

```
String[][] names = {
            {"Mr.", "Mrs.", "Ms."},
            {"John", "Gupta", "Hegde", "Khan"},
            {":M", ":F"}
          };
System.out.println(names[0][2] + names[1][2]+ names[2][1]);
```

1). Compile error

2). Mrs.Gupta : M

3). **Ms.Hegde : F**

4). Mr.Khan : M

Solution :
option [3] is correct

11. Q. What is the output?

```
public class Test {
   public static void main(String[] args) {
      StringBuilder a = new StringBuilder("A");
      StringBuilder b = new StringBuilder("B");
      change(a, b);
      System.out.println(a + "," + b);
   }
   static void change(StringBuilder x, StringBuilder y) {
      y.append(x);
      y = x;
   }
}
```

1). **A,BA**

2). A,B

3). A,A

4). compile Erorr

Solution :
option [1] is correc

12. Q. What will be the output?

```
public class Test_1 {
   static String str;
      public static void main(String[] args) {
       String s;
       if(str=="abc")
          s = str+10;
       System.out.println(str);
       System.out.print(s);
      }
}
```

1). null and null are printed as output.

2). runtime error.

3). Compile error, cannot access str from main.

4). **Compile error, s not initialised.**

Solution :
option [4] is correct

13. Q. What will be the result?

```
class TestModifiers{
        int i;
        public static void main (String[] args) {
                 int i; //1
                private int a = 1; //2
                protected int b = 1; //3
                public int c = 1; //4
                System.out.println(a+b+c); //5
        }
}
```

1). compiletime error at line 1,2,3,4,5;

2). **compiletime error at line 2,3,4,5;**

3). compiletime error at line 2,3,4;

4). Prints 3

5). None of the above

Solution :
option [2] is correct

14. Q. Which two classes inherit the Shape class correctly?

```
public abstract class Shape {
private int x;
private int y;
public abstract void draw();
public void setAnchor(int x, int y)
{
this.x = x; this.y = y;
}
            }
```

1). public class Circle implements Shape { private int radius; }

2). **public abstract class Circle extends Shape { private int radius; }**

3). public class Circle extends Shape { private int radius; public void draw(); }

4). public abstract class Circle implements Shape { private int radius; public void draw(); }

5). **public class Circle extends Shape {**
private int radius; public void draw() {/* code here */} }

Solution :
option [2,5] are correct

15. Q. Which of the following is true?

A. An interface cannot be instantiated.
B. A final field of a class can be instantiated in the constructor.
C. A protected field of a class is accessible by child class in any package.
D. A class can implement more than one interface

1). A & D

2). A, C & D

3). A, B &D

4). **All of them**

Solution :
option [4] is correct

16. Q. Given:

```
1. static void test() throws RuntimeException {
2. try {
3. System.out.print("test ");
4. throw new RuntimeException();
5. }
6. catch (Exception ex) { System.out.print("exception "); }
7. }
8. public static void main(String[] args) {
9. try { test(); }
10. catch (RuntimeException ex) { System.out.print("runtime "); }
```

```
11. System.out.print("end ");
12. }
```

What will be the result?

1). test ends

2). compilation error

3). test runtime end

4). **test exception end**

5). A Throwable is thrown by main at runtime.

Solution :
option [4] is correct

17. Q. The following code :

```
public class Test {
   public static void main(String[] args) {
   int a=5, b=7;
   if(a & b > 0 && a | b > 0)
      System.out.println("true");
   else
      System.out.println("false");
   }
}
```

1). prints output true

2). No output.

3). **does not compile**

4). **runtime exception**

Solution :
option [3] is correct

18.**Q.** What will be the result?

```
public class Yippee {
      public static void main(String [] args) {
            for(int x = 1; x < args.length; x++) {
                  System.out.print(args[x] + " ");
            }
      }
}
```

and two separate command line invocations:

```
      java Yippee
      java Yippee 1 2 3 4
```

1). No output is produced. 1 2 3

2). **No output is produced. 2 3 4**

3). No output is produced. 1 2 3 4

4). An exception is thrown at runtime. 1 2 3

5). An exception is thrown at runtime. 2 3 4

6). An exception is thrown at runtime. 1 2 3 4

Solution :
option [2] is correct

19. Q. Which statement is true?

1). A class's finalize() method CANNOT be invoked explicitly.

2). super.finalize() is called implicitly by any overriding finalize() method.

3). **The finalize() method for a given object is called no more than once by the garbage collector.**

4). The order in which finalize() is called on two objects is based on the order in which the two objects
became finalizable.

Solution :
option [3] is correct

20. Q. The following code :

```
public class Test{
  public static void main(String args[]) {
    B b = new B();
  }
}
class A {
  A() {
    System.out.print("A");
  }
}
class B extends A{
  B() {
    System.out.print("B");
  }
}
```

1). Gives output : BA

2). **Gives output : AB**

3). Gives output : B

4). Compilation Error

Solution :
option [2] is correct