1.

What will happen when the following program is compiled and run?

```
public class SM
{
    public String checkIt(String s)
    {
        if(s.length() == 0 || s == null)
        {
            return "EMPTY";
        }
        else return "NOT EMPTY";
    }

    public static void main(String[] args)
    {
        SM a = new SM();
        a.checkIt(null);
    }
}
```

Please select 1 option

○ It will print EMPTY.

○ It will print NOT EMPTY.

○ It will throw NullPointerException.

○ It will print EMPTY if || is replaced with |.

2.

A try statement must always have a .... associated with it.

Please select 1 option

○ catch

○ throws

○ finally

○ catch, finally or both

○ throw

3.

What will be the result of compiling and running the following program ?

```
public class ExceptionTest
{
    public static void main(String[] args) throws Exception
    {
        try
        {
            m2();
        }
        finally
        {
            m3();
        }
        catch (Exception e){
        }
    }

    public static void m2() throws Exception { throw new Exception("from m2"); }

    public static void m3() throws Exception{ throw new Exception("from m3"); }
}
```

○ It will print an exception stack trace with message `from m2` when run.

○ It will print an exception stack trace with message "from m3" when run.

○ It will not print anything when run.

○ It will not compile.

○ None of the above.

4.

What letters, and in what order, will be printed when the following program is compiled and run?

```
public class FinallyTest
{
    public static void main(String args[]) throws Exception
    {
        try
        {
            m1();
            System.out.println("A");
        }
        finally
        {
            System.out.println("B");
        }
        System.out.println("C");
    }
    public static void m1() throws Exception { throw new Exception(); }
}
```

Please select 1 option

○ It will print C and B, in that order.

○ It will print A and B, in that order.

○ It will print B and then throw an Exception.

○ It will print A, B and C in that order.

○ Compile time error.

5.

Given:

```
class Node{
    int id;
    Node node;
    public static void main(String[] args) {
        Node n =  new Node();
        System.out.println(n.id);
        System.out.println(n.node.id);
    }
}
```

What is the result?

Please select 1 option

○ 0
   0

○ 0
   null

○ 0
   exception at runtime

○ Compilation failure

6.

Which code fragments will print the last argument given on the command line to the standard output, and exit without any output and exceptions on the command line if no arguments are given?

Please select 3 options

```
public static void main(String args[ ])
{
    if (args.length != 0)   System.out.println(args[args.length-1]);
}
```

```
public static void main(String args[ ])
{
    try {      System.out.println(args[args.length-1]);         }
    catch (ArrayIndexOutOfBoundsException e) {     }
}
```

```
public static void main(String args[ ])
{
    int i = args.length;
    if (i != 0) System.out.println(args[i-1]);
}
```

```
public static void main(String args[ ])
{
    int i = args.length-1;
    if (i > 0) System.out.println(args[i]);
}
```

```
public static void main(String args[ ])
{
    try { System.out.println(args[args.length-1]); }
    catch (NullPointerException e) {}
}
```

7.

What will be the ouput of the following program:

```
public class TestClass
{
    public static void main(String args[])
    {
        try{
            m1();
        }catch(IndexOutOfBoundsException e){
            System.out.println("1");
            throw new NullPointerException();
        }catch(NullPointerException e){
            System.out.println("2");
            return;
        }catch (Exception e) {
            System.out.println("3");
        }finally{
            System.out.println("4");
        }
        System.out.println("END");
    }
    // IndexOutOfBoundsException is a subclass of RuntimeException.
    static void m1()
    {
        System.out.println("m1 Starts");
        throw new IndexOutOfBoundsException( "Big Bang " );
    }
}
```

☐ The program will print 'm1 Starts'.

☐ The program will print 'm1 Starts', 1 and 4, in that order.

☐ The program will print 'm1 Starts', 1, 2 in that order.

☐ The program will print 'm1 Starts', 1, 2 and 4 in that order.

☐ 'END' will not be printed.

8.

Given the following program, which of these statements are true?

```java
public class FinallyTest
{
    public static void main(String args[])
    {
        try
        {
            if (args.length == 0) return;
            else throw new Exception("Some Exception");
        }
        catch(Exception e)
        {
            System.out.println("Exception in Main");
        }
        finally
        {
            System.out.println("The end");
        }
    }
}
```

Please select 2 options

☐ If run with no arguments, the program will only print The end.

☐ If run with one argument, the program will only print The end.

☐ If run with one argument, the program will print Exception in Main and The end.

☐ If run with one argument, the program will only print Exception in Main.

☐ Only one of the above is correct.

9.

Given:

```java
public class Test
{
    public int div(int a, int b) throws Exception {
        try{
            return a/b;
        }catch(ArithmeticException ae){
            System.out.println("exception in div");
            return 0;
        }
    }

    public static void main(String args[])
    {
        Test test = new Test();
        try{
            System.out.println(test.div(5, 0));
        }catch(Exception e){
            System.out.println("exception in main");
        }
    }
}
```

What is the output?

○ exception in div
exception in main

○ exception in div

○ exception in main

○ exception in div
0

○ Compilation failure

10.

What will be the output of the following class...

```
class Test
{
   public static void main(String[] args)
   {
      int j = 1;
      try
      {
         int i = doIt() / (j = 2);
      } catch (Exception e)
      {
         System.out.println(" j = " + j);
      }
   }
   public static int doIt() throws Exception {   throw new Exception("FORGET IT");   }
}
```

Please select 1 option

○ It will print j = 1;

○ It will print j = 2;

○ The value of j cannot be determined.

○ It will not compile.

○ None of the above.

11.

Which digits and in what order will be printed when the following program is run?

```
public class TestClass
{
   public static void main(String args[])
   {
      int k = 0;
      try{
         int i = 5/k;
      }
      catch (ArithmeticException e){
         System.out.println("1");
      }
      catch (RuntimeException e){
         System.out.println("2");
         return ;
      }
      catch (Exception e){
         System.out.println("3");
      }
      finally{
         System.out.println("4");
      }
      System.out.println("5");
   }
}
```

Please select 1 option

○ The program will print 5.

○ The program will print 1 and 4, in that order.

○ The program will print 1, 2 and 4, in that order.

○ The program will print 1, 4 and 5, in that order.

○ The program will print 1,2, 4 and 5, in that order.

12.

What will be the output of the following program?

```java
public class TestClass
{
    public static void main(String[] args) throws Exception
    {
        try{
            amethod();
            System.out.println("try");
        }
        catch(Exception e){
            System.out.println("catch");
        }
        finally {
            System.out.println("finally");
        }
        System.out.println("out");
    }

public static void amethod(){ }

}
```

Please select 1 option

○ try finally

○ try finally out

○ try out

○ catch finally out

○ It will not compile because amethod() does not throw any exception.

13.

What will the following program print?

```java
public class TestClass
{
    public static void main(String[] args)
    {
        int x = 1;
        int y = 0;
        if( x/y ) System.out.println("Good");
        else   System.out.println("Bad");
    }
}
```

Please select 1 option

○ Good

○ Bad

○ Exception at runtime saying division by Zero.

○ It will not compile.

○ None of the above.

14.

What will be the result of attempting to compile and run the following program?

```
class TestClass
{
    public static void main(String args[])
    {
        int i = 0;
        loop :          // 1
        {
            System.out.println("Loop Lable line");
            try
            {
                for (  ;  true ;  i++ )
                {
                    if( i >5) break loop;        // 2
                }
            }
            catch(Exception e)
            {
                System.out.println("Exception in loop.");
            }
            finally
            {
                System.out.println("In Finally");      // 3
            }
        }
    }
}
```

Please select 1 option

○ Compilation error at line 1 as this is an invalid syntax for defining a label.

○ Compilation error at line 2 as 'loop' is not visible here.

○ No compilation error and line 3 will be executed.

○ No compilation error and line 3 will NOT be executed.

○ Only the line with the label Loop will be printed.

15.

Consider the following code ...

```
class A
{
    public void doA(int k) throws Exception {  // 0
        for(int i=0; i< 10; i++) {
            if(i == k) throw new Exception("Index of k is "+i); // 1
        }
    }
    public void doB(boolean f) { //2
        if(f) {
            doA(15); //3
        }
        else return;
    }
    public static void main(String[] args) { //4
        A a = new A();
        a.doB(args.length>0); //5
    }
}
```
Which of the following statements are correct?

Please select 1 option

○ This will compile and run without any errors or exception.

○ This will compile if 'throws Exception' is added at line //2

○ This will compile if 'throws Exception' is added at line //4

○ This will compile if 'throws Exception' is added at line //2 as well as //4

○ This will compile if line marked // 1 is enclosed in a try - catch block.

16.

Identify the exceptions that will be received when the code snippets on the left hand side are executed.

1.
```
int factorial(int n){
 if(n==1) return 1;
 else return n*factorial(n-1);
}
```
Assume that it is called with a very big integer.

2.
```
void printMe(Object[] oa){
 for(int i=0; i<=oa.length; i++)
   System.out.println(oa[i]);
}
```
Assume that it is called as such: printMe(null);

3.
```
Object m1( ){
 return new Object();
}
void m2( ){
 String s = (String) m1( );
}
```

ClassCastException   **ArrayIndexOutOfBoundsException**

StackOverflowError   **NullPointerException**

No Exception Will Be Thrown.   **Will Not Compile.**