

What will be the output when the following program is run?

```
package exceptions;
public class TestClass{
    public static void main(String[] args) {
        try{
            hello();
        }
        catch(MyException me){
            System.out.println(me);
        }
    }

    static void hello() throws MyException{
        int[] dear = new int[7];
        dear[0] = 747;
        foo();
    }

    static void foo() throws MyException{
        throw new MyException("Exception from foo");
    }
}

class MyException extends Exception {
    public MyException(String msg){
        super(msg);
    }
}
```

(Assume that line numbers printed in the messages given below are correct.)

Please select 1 option

- Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 10  
at exceptions.TestClass.doTest(TestClass.java:24)  
at exceptions.TestClass.main(TestClass.java:14)
- Error in thread "main" java.lang.ArrayIndexOutOfBoundsException
- exceptions.MyException: Exception from foo  
  
exceptions.MyException: Exception from foo  
at exceptions.TestClass.foo(TestClass.java:29)  
at exceptions.TestClass.hello(TestClass.java:25)  
at exceptions.TestClass.main(TestClass.java:14)

Which of the lines will cause a compile time error in the following program?

```
public class MyClass{
    public static void main(String args[]){
        char c;
        int i;
        c = 'a';//1
        i = c; //2
        i++; //3
        c = i; //4
        c++; //5
    }
}
```

Please select 1 option

line 1

line 2

line 3

line 4

line 5

What will the following program print?

```
public class InitTest{
    public InitTest(){
        s1 = sM1("1");
    }
    static String s1 = sM1("a");
    String s3 = sM1("2");
    {
        s1 = sM1("3");
    }
    static{
        s1 = sM1("b");
    }
    static String s2 = sM1("c");
    String s4 = sM1("4");
    public static void main(String args[]){
        InitTest it = new InitTest();
    }
    private static String sM1(String s){
        System.out.println(s);  return s;
    }
}
```

Please select 1 option

- The program will not compile.
- It will print : a b c 2 3 4 1
- It will print : 2 3 4 1 a b c
- It will print : 1 a 2 3 b c 4
- It will print : 1 a b c 2 3 4

Consider the following code for the main() method:

```
public static void main(String[] args) throws Exception{
    int i = 1, j = 10;
    do {
        if (i++ > --j) continue;
    } while (i < 5);
    System.out.println("i=" + i + " j=" + j);
}
```

What will be the output when the above code is executed?

Please select 1 option

i=6 j=6

i=5 j=6

i=5 j=5

i=6 j=5

None of these.

Given:

```
String mStr = "123";
```

```
long m = // 1
```

Which of the following options when put at //1 will assign 123 to m?

Please select 3 options

- new Long(mStr);
- Long.parseLong(mStr);
- Long.longValue(mStr);
- (new Long()).parseLong(mStr);
- Long.valueOf(mStr).longValue();

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

```
public class TestClass{  
    public static void main(String args[ ] ){  
        int i, j, k;  
        i = j = k = 9;  
        System.out.println(i);  
    }  
}
```

Please select 2 options

- The code will not compile because unlike in c++, operator '=' cannot be chained i.e. a = b = c = d is invalid.
- The code will not compile as 'j' is being used before getting initialized.
- The code will compile correctly and will display '9' when run.
- The code will not compile as 'j' and 'i' are being used before getting initialized.
- All the variables will get a value of 9.

What will the following code print when run?

```
public class TestClass {  
    public void switchString(String input){  
        switch(input){  
            case "a" : System.out.println( "apple" );  
            case "b" : System.out.println( "bat" );  
                break;  
            case "B" : System.out.println( "big bat" );  
            default : System.out.println( "none" );  
        }  
    }  
  
    public static void main(String[] args) throws Exception {  
        TestClass tc = new TestClass();  
        tc.switchString("B");  
    }  
}
```

Please select 1 option

- bat
- big bat
- big bat
- none
- big bat
- bat
- The code will not compile.

Identify the valid code fragments when occurring by themselves within a method.

---

Please select 1 option

- long y = 123\_456\_L;
  - long z = \_123\_456L;
  - float f1 = 123\_.345\_667F;
  - float f2 = 123\_345\_667F;
  - None of the above declarations are valid.
-

What will be the contents of s1 and s2 at the time of the `println` statement in the main method of the following program?

```
import java.util.*;
public class TestClass{
    public static void main(String args[]){
        Stack s1 = new Stack ();
        Stack s2 = new Stack ();
        processStacks (s1,s2);
        System.out.println (s1 + "    " + s2);
    }
    public static void processStacks(Stack x1, Stack x2){
        x1.push (new Integer ("100")); //assume that the method push adds the passed object to the stack.
        x2 = x1;
    }
}
```

Note:[] is used in the options below to denote each element of the Stack.

---

Please select 1 option

[100] [100]

---

[100] []

---

[] [100]

---

[] []

---

Which of the following are valid at line 1?

```
public class X{  
    //line 1: insert code here.  
}
```

Please select 2 options

String s;

String s = 'asdf';

String s = 'a';

String s = this.toString();

String s = asdf;

Given:

```
package strings;
public class StringFromChar {

    public static void main(String[] args) {
        String myStr = "good";
        char[] myCharArr = {'g', 'o', 'o', 'd' };

        String newStr = null;
        for(char ch : myCharArr){
            newStr = newStr + ch;
        }

        System.out.println((newStr == myStr)+ " " + (newStr.equals(myStr)));
    }
}
```

What will it print when compiled and run?

Please select 1 option

true true

true false

false true

false false

What will be the output when the following program is run?

```
public class TestClass{  
    public static void main(String args[]){  
        int i;  
        int j;  
        for (i = 0, j = 0; j < i; ++j, i++){  
            System.out.println(i + " " + j);  
        }  
        System.out.println(i + " " + j);  
    }  
}
```

Please select 1 option

- 0 0 will be printed twice.
- 0 0 will be printed once.
- It will keep on printing 0 0
- It will not compile.
- It will print 0 0 and then 0 1.

Consider following classes:

```
//In File Other.java
package other;
public class Other { public static String hello = "Hello"; }

//In File Test.java
package testPackage;
import other.*;
class Test{
    public static void main(String[] args){
        String hello = "Hello", lo = "lo";
        System.out.print((testPackage.Other.hello == hello) + " "); //line 1
        System.out.print((other.Other.hello == hello) + " "); //line 2
        System.out.print((hello == ("Hel"+lo)) + " "); //line 3
        System.out.print((hello == ("Hel"+lo)) + " "); //line 4
        System.out.println(hello == ("Hel"+lo).intern()); //line 5
    }
}
class Other { static String hello = "Hello"; }
```

What will be the output of running class Test?

---

Please select 1 option

false false true false true

---

false true true false true

---

true true true true true

---

true true true false true

---

None of the above.

Given the following code :

```
public class TestClass {

    int[][] matrix = new int[2][3];

    int a[] = {1, 2, 3};
    int b[] = {4, 5, 6};

    public int compute(int x, int y){
        //1 : Insert Line of Code here
    }

    public void loadMatrix(){
        for(int x=0; x<matrix.length; x++){
            for(int y=0; y<matrix[x].length; y++){
                //2: Insert Line of Code here
            }
        }
    }
}
```

What can be inserted at //1 and //2?

Please select 1 option

- return a(x)\*b(y);
- and  
matrix(x, y) = compute(x, y);
- return a[x]\*b[y];
- and  
matrix[x, y] = compute(x, y);
- return a[x]\*b[y];
- and  
matrix[x][y] = compute(x, y);
- return a(x)\*b(y);
- and  
matrix(x)(y) = compute(x, y);
- return a[x]\*b[y];
- and  
matrix[[x][y]] = compute(x, y);

---

What will the following code print when compiled and run?

```
public class OrderTest {  
  
    public void initData(String[] arr){  
        int ind = 0;  
        for(String str : arr){  
            str.concat(str+" "+ind);  
            ind++;  
        }  
    }  
  
    public void printData(String[] arr){  
        for(String str : arr){  
            System.out.println(str);  
        }  
    }  
  
    public static void main(String[] args) {  
        OrderTest ot = new OrderTest();  
        String[] arr = new String[2];  
        ot.initData(arr);  
        ot.printData(arr);  
    }  
}
```

---

Please select 1 option

null 0  
 null 1

---

0  
 1

---

0  
 1

---

(There is a space before 0 and 1)

---

null  
 null

---

It will throw a RuntimeException at run time.

---

What will the following code print?

```
public class Test{
    public static void testInts(Integer obj, int var){
        obj = var++;
        obj++;
    }
    public static void main(String[] args) {
        Integer val1 = new Integer(5);
        int val2 = 9;
        testInts(val1++, ++val2);
        System.out.println(val1+" "+val2);
    }
}
```

Please select 1 option

10 9

10 10

6 9

6 10

5 11

Consider the following method...

```
public int setVar(int a, int b, float c) { ...}
```

Which of the following methods correctly overload the above method?

---

Please select 2 options

public int setVar(int a, float b, int c){  
 return (int)(a + b + c);  
}

---

public int setVar(int a, float b, int c){  
 return this(a, c, b);  
}

---

public int setVar(int x, int y, float z){  
 return x+y;  
}

---

public float setVar(int a, int b, float c){  
 return c\*a;  
}

---

public float setVar(int a){  
 return a;  
}

---

Consider the following interface definition:

```
interface Bozo{
    int type = 0;
    public void jump();
}
```

Now consider the following class:

```
public class Type1Bozo implements Bozo{
    public Type1Bozo(){
        type = 1;
    }

    public void jump(){
        System.out.println("jumping..."+type);
    }

    public static void main(String[] args){
        Bozo b = new Type1Bozo();
        b.jump();
    }
}
```

What will the program print when compiled and run?

---

Please select 1 option

jumping...0

---

jumping...1

---

This program will not compile.

---

It will throw an exception at runtime.

---

Consider the contents of following two files:

```
//In file A.java
package a;
public class A{
    A(){}
    public void print(){ System.out.println("A"); }
}

//In file B.java
package b;
import a.*;
public class B extends A{
    B(){}
    public void print(){ System.out.println("B"); }
    public static void main(String[] args){
        new B();
    }
}
```

What will be printed when you try to compile and run class B?

---

Please select 1 option

- It will print A.
  - It will print B.
  - It will not compile.
  - It will compile but will not run.
  - None of the above.
-

What will the following code print?

```
String abc = "";  
abc.concat("abc");  
abc.concat("def");  
System.out.print(abc);
```

---

Please select 1 option

abc

---

abcdef

---

def

---

It will print empty string (or in other words, nothing).

---

It will not compile because there is no concat() method in String class.

---

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

```
class Base{
    public Object getValue(){ return new Object(); } //1
}

class Base2 extends Base{
    public String getValue(){ return "hello"; } //2
}

public class TestClass{
    public static void main(String[] args){
        Base b = new Base2();
        System.out.println(b.getValue()); //3
    }
}
```

---

Please select 1 option



It will print the hash code of the object.

---



It will print hello.

---



Compile time error at //1.

---



Compile time error at //2.

---



Compile time error at //3.

---

Which of the following standard java exception classes extend `java.lang.RuntimeException`?

Please select 4 options

- `java.lang.SecurityException`
- `java.lang.ClassCastException`
- `java.lang.NullPointerException`
- `java.lang.CloneNotSupportedException`
- `java.lang.IndexOutOfBoundsException`

What will the following code print when compiled and run?

```
class ABCD{  
    int x = 10;  
    static int y = 20;  
}  
class MNOP extends ABCD{  
    int x = 30;  
    static int y = 40;  
}  
  
public class TestClass {  
    public static void main(String[] args) {  
        System.out.println(new MNOP().x+", "+new MNOP().y);  
    }  
}
```

Please select 1 option

10, 40

30, 20

10, 20

30, 40

20, 30

Compilation error.

The following code snippet will print true.

```
String str1 = "one";
String str2 = "two";
System.out.println( str1.equals(str1=str2) );
```

Please select 1 option

True

False

What will the following code print when run?

```
public class TestClass {  
    public static void main(String[] args) throws Exception {  
        String[] sa = {"a", "b", "c"};  
        for(String s : sa){  
            if("b".equals(s)) continue;  
            System.out.println(s);  
            if("b".equals(s)) break;  
            System.out.println(s+" again");  
        }  
    }  
}
```

Please select 1 option

- a
- a again
- c
- c again

- 
- a
  - a again
  - b

- 
- a
  - a again
  - b
  - b again

- 
- c
  - c again

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 throw Exception.
- It will print A, B and C in that order.
- Compile time error.

Given:

```
public class TestClass{  
    public static int getSwitch(String str){  
        return (int) Math.round( Double.parseDouble(str.substring(1, str.length()-1)) );  
    }  
    public static void main(String args []){  
        switch(getSwitch(args[0])){  
            case 0 : System.out.print("Hello ");  
            case 1 : System.out.print("World"); break;  
            default : System.out.print("Good Bye");  
        }  
    }  
}
```

What will be printed by the above code if it is run with command line:

java TestClass --0.50

(There are two minuses before 0.)

---

Please select 1 option

Hello

---

World

---

Hello World

---

Hello World Good Bye

---

Good Bye

---

What will the following code print when compiled and run?

```
abstract class Calculator{  
    abstract void calculate();  
    public static void main(String[] args){  
        System.out.println("calculating");  
        Calculator x = null;  
        x.calculate();  
    }  
}
```

Please select 1 option

- It will not compile.
- It will not print anything and will throw NullPointerException
- It will print calculating and then throw NullPointerException.
- It will print calculating and will throw NoSuchMethodError
- It will print calculating and will throw MethodNotImplementedException

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

```
public class TestClass{  
    public static void main(String args[ ] ){  
        A o1 = new C( );  
        B o2 = (B) o1;  
        System.out.println(o1.m1( ) );  
        System.out.println(o2.i );  
    }  
}  
class A { int i = 10;  int m1( ) { return i; } }  
class B extends A { int i = 20;  int m1() { return i; } }  
class C extends B { int i = 30;  int m1() { return i; } }
```

---

Please select 1 option

The program will fail to compile.

---

Class cast exception at runtime.

---

It will print 30, 20.

---

It will print 30, 30.

---

It will print 20, 20.

---

The following class will print 'index = 2' when compiled and run.

```
class Test{  
    public static int[ ] getArray() { return null; }  
    public static void main(String[] args){  
        int index = 1;  
        try{  
            getArray()[index=2]++;  
        }  
        catch (Exception e){ } //empty catch  
        System.out.println("index = " + index);  
    }  
}
```

---

Please select 1 option

True

---

False

What is the result of executing the following fragment of code:

```
boolean b1 = false;  
boolean b2 = false;  
if (b2 != b1 = !b2){  
    System.out.println("true");  
}  
else{  
    System.out.println("false");  
}
```

---

Please select 1 option

Compile time error.

---

It will print true.

---

It will print false.

---

Runtime error.

---

It will print nothing.

---

Given the following class, which of the given blocks can be inserted at line 1 without errors?

```
public class InitClass{  
    private static int loop = 15 ;  
    static final int INTERVAL = 10 ;  
    boolean flag ;  
    //line 1  
}
```

---

Please select 4 options

static {System.out.println("Static"); }

static { loop = 1; }

static { loop += INTERVAL; }

static { INTERVAL = 10; }

{ flag = true; loop = 0; }

---

What will the following code print when compiled and run?

```
class StringWrapper {  
    private String theVal;  
    public StringWrapper(String str){ this.theVal = str; }  
}  
public class Tester{  
    public static void main(String[] args) {  
        StringWrapper sw = new StringWrapper("How are you?");  
        StringBuilder sb = new StringBuilder("How are you?");  
        System.out.println("Hello, "+sw);  
        System.out.println("Hello, "+sb);  
    }  
}
```

---

Please select 1 option

Hello, How are you?  
 Hello, How are you?

---

Hello, StringWrapper@<hashcode>  
 Hello, How are you?

---

Hello, How are you?  
 Hello, StringBuilder@<hashcode>

---

Hello, How are you?  
 Hello, java.lang.StringBuilder@<hashcode>

---

Hello, StringWrapper@<hashcode>  
 Hello, java.lang.StringBuilder@<hashcode>

---

Which of the following is a legal return type of a method overriding the given method:

```
public Object myMethod() {...}
```

(Select the best option.)

Please select 1 option

Object

String

Return type can be any class since all objects can be cast to Object.

void

None of the above.

Consider the following class and interface definitions (in separate files):

```
public class Sample implements IInt{
    public static void main(String[] args){
        Sample s = new Sample(); //1
        int j = s.thevalue; //2
        int k = IInt.thevalue; //3
        int l = thevalue; //4
    }
}
public interface IInt{
    int thevalue = 0;
}
```

What will happen when the above code is compiled and run?

---

Please select 1 option

It will give an error at compile time at line //1.

---

It will give an error at compile time at line //2.

---

It will give an error at compile time at line //3

---

It will give an error at compile time at line //4.

---

It will compile and run without any problem.

---

What will the following code print when run?

```
public class TestClass{  
    public static long main(String[] args){  
        System.out.println("Hello");  
        return 10L;  
    }  
}
```

---

Please select 1 option

Hello

---

It will not print anything.

---

It will not compile

---

It will throw an Error at runtime.

---

None of the above.

---

In which sequence will the characters a, b, and, c be printed by the following program?

```
import java.util.* ;
public class ListTest{
    public static void main(String args[]){
        List s1 = new ArrayList( );
        s1.add("a");
        s1.add("b");
        s1.add(1, "c");
        List s2 = new ArrayList( s1.subList(1, 1) );
        s1.addAll(s2);
        System.out.println(s1);
    }
}
```

---

Please select 1 option

- The sequence a, b, c is printed.
  - The sequence a, b, c, b is printed.
  - The sequence a, c, b, c is printed.
  - The sequence a, c, b is printed.
  - None of the above.
-

Consider the following program...

```
class Super { }
class Sub extends Super { }
public class TestClass{
    public static void main(String[] args){
        Super s1 = new Super(); //1
        Sub s2 = new Sub();     //2
        s1 = (Super) s2;       //3
    }
}
```

Which of the following statements are correct?

Please select 1 option

- It will compile and run without any problems.
- It will compile but WILL throw ClassCastException at runtime.
- It will compile but MAY throw ClassCastException at runtime.
- It will not compile.
- None of the above.

Consider the following two classes defined in two .java files.

```
//in file /root/com/foo/X.java
package com.foo;
public class X{
    public static int LOGICID = 10;
    public void apply(int i){
        System.out.println("applied");
    }
}
```

```
//in file /root/com/bar/Y.java
package com.bar;
//1 <== INSERT STATEMENT(s) HERE
public class Y{
    public static void main(String[] args){
        System.out.println(X.LOGICID);
    }
}
```

What should be inserted at //1 so that Y.java can compile without any error?

---

Please select 1 option

- import static X;
  - import static com.foo.\*;
  - import static com.foo.X.\*;
  - import com.foo.\*;
  - import com.foo.X.LOGICID;
-

---

What will the following code print when run?

```
class A {  
}
```

```
class AA extends A {  
}
```

```
public class TestClass {  
    public static void main(String[] args) throws Exception {  
        A a = new A();  
        AA aa = new AA();  
        a = aa;  
        System.out.println("a = "+a.getClass());  
        System.out.println("aa = "+aa.getClass());  
    }  
}
```

---

Please select 1 option

---

It will not compile.

---

It will throw ClassCastException at runtime.

---

a = class AA  
aa = class AA

---

a = class A  
aa = class AA

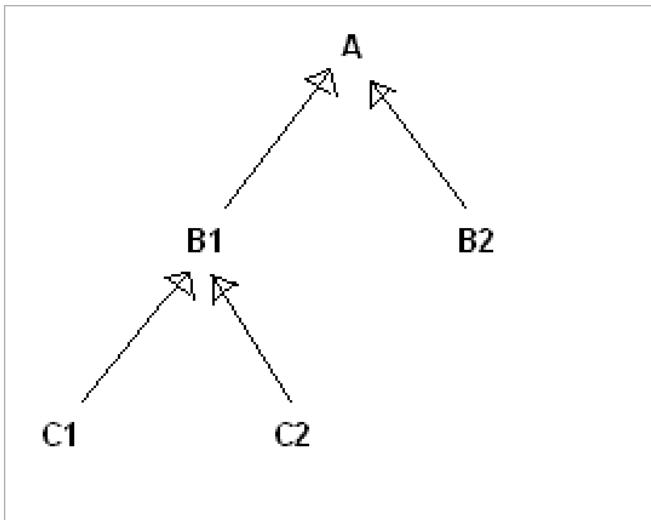
---

Consider the following class hierarchy shown in the image. (B1 and B2 are subclasses of A and C1, C2 are subclasses of B1)

Assume that method `public void m1(){ ... }` is defined in all of these classes EXCEPT B1 and C1.

Assume that "objectOfXX" means a variable that points to an object of class XX. So, `objectOfC1` means a reference variable that is pointing to an object of class C1.

Which of the following statements are correct?



Please select 1 option

- `objectOfC1.m1();` will cause a compilation error.
- `objectOfC2.m1();` will cause A's `m1()` to be called.
- `objectOfC1.m1();` will cause A's `m1()` to be called.
- `objectOfB1.m1();` will cause an exception at runtime.
- `objectOfB2.m1();` will cause an exception at runtime.

What will the following code snippet print?

```
Object t = new Integer(107);
int k = (Integer) t.intValue()/9;
System.out.println(k);
```

---

Please select 1 option

11

---

12

---

It will not compile.

---

It will throw an exception at runtime.

---

What can be inserted in the code below so that it will print true when run?

```
public class TestClass{  
  
    public static boolean checkList(List list, Predicate<List> p){  
        return p.test(list);  
    }  
  
    public static void main(String[] args) {  
        boolean b = //WRITE CODE HERE  
        System.out.println(b);  
    }  
}
```

Please select 2 options

- `checkList(new ArrayList(), al -> al.isEmpty());`
- `checkList(new ArrayList(), ArrayList al -> al.isEmpty());`
- `checkList(new ArrayList(), al -> return al.size() == 0);`
- `checkList(new ArrayList(), al -> al.add("hello"));`
- `checkList(new ArrayList(), (ArrayList al) -> al.isEmpty());`

Given the following code -

```
public class MyFirstClass{  
    public static void main(String[] args){  
        System.out.println(args[1]);  
    }  
}
```

Which of the following commands will compile and then print "hello"?

---

Please select 1 option

- javac MyFirstClass  
java MyFirstClass hello hello
  - javac MyFirstClass.java  
java MyFirstClass hello hello
  - javac MyFirstClass  
java MyFirstClass hello
  - javac MyFirstClass.java  
java MyFirstClass hello
- 
- .....

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

```
class Base{
    public short getValue(){ return 1; } //1
}
class Base2 extends Base{
    public byte getValue(){ return 2; } //2
}
public class TestClass{
    public static void main(String[] args){
        Base b = new Base2();
        System.out.println(b.getValue()); //3
    }
}
```

---

Please select 1 option

It will print 1

---

It will print 2.

---

Compile time error at //1

---

Compile time error at //2

---

Compile time error at //3

---

You have a method that currently does not handle any exception thrown from the code contained in its method body. You are now changing this method to call another method that throws IOException.

What changes, independent of each other, can you make to your method so that it will compile?

---

Please select 2 options

- Set the exception to null and don't rethrow it.
  - Declare IOException in the throws clause of your method.
  - Wrap the call to another method within a try-catch block that catches RuntimeException.
  - Wrap the call to another method within a try-catch block that catches Exception.
-

What will be the output of the following program?

```
public class TestClass{  
    public static void main(String args[ ] ){  
        int i = 0 ;  
        boolean bool1 = true ;  
        boolean bool2 = false;  
        boolean bool = false;  
        bool = ( bool2 & method1(i++) ); //1  
        bool = ( bool2 && method1(i++) ); //2  
        bool = ( bool1 | method1(i++) ); //3  
        bool = ( bool1 || method1(i++) ); //4  
        System.out.println(i);  
    }  
    public static boolean method1(int i){  
        return i>0 ? true : false;  
    }  
}
```

Please select 1 option

It will print 1.

It will print 2.

It will print 3.

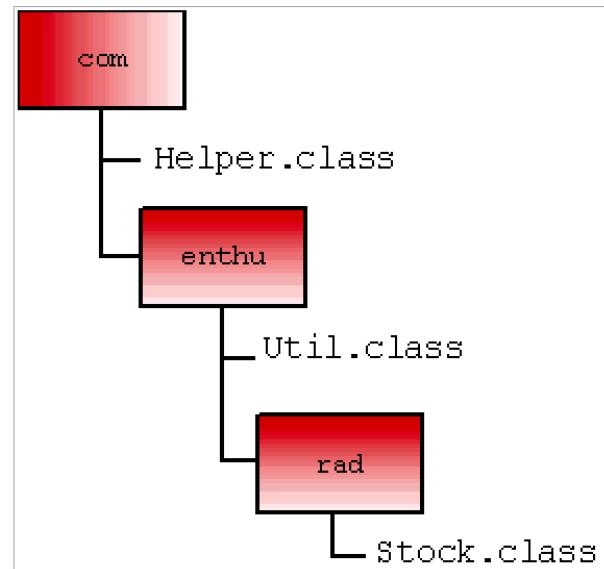
It will print 4.

It will print 0.

Consider the directory structure shown in Image 1 that displays available folders and classes and the code given below:

```
class StockQuote{
    Stock stock;
    public StockQuote(Stock s)  {
    }
    public void store() throws IOException{
        Util.store(stock);
    }
    public double computePrice(){
        return Helper.getPricer(stock).price();
    }
}
```

Assuming that the code uses valid method calls, what statements MUST be added to the above class?



Please select 4 options

- package com.enthu.rad.\*;
- import com.enthu.\*;
- package com.enthu.rad;
- import com.\*;
- import java.io.\*;
- It is not required to import java.io.\* or import java.io.IOException because java.io package is imported automatically.

Which line will print the string "MUM"?

```
public class TestClass{  
    public static void main(String args []){  
        String s = "MINIMUM";  
        System.out.println(s.substring(4, 7)); //1  
        System.out.println(s.substring(5)); //2  
        System.out.println(s.substring(s.indexOf('I', 3))); //3  
        System.out.println(s.substring(s.indexOf('I', 4))); //4  
    }  
}
```

---

Please select 1 option

1

---

2

---

3

---

4

---

None of these.

---

Which of the following statements are true?

---

Please select 2 options

- method length() of String class is a final method.

---

  - You can make mutable subclasses of the String class.

---

  - StringBuilder extends String.

---

  - StringBuilder is a final class.

---

  - String class is not final.
-

What, if anything, is wrong with the following code?

```
interface T1{  
}  
interface T2{  
    int VALUE = 10;  
    void m1();  
}  
  
interface T3 extends T1, T2{  
    public void m1();  
    public void m1(int x);  
}
```

---

Please select 1 option

- T3 cannot implement both T1 and T2 because it leads to ambiguity.
  - There is nothing wrong with the code.
  - The code will work fine only if VALUE is removed from T2 interface.
  - The code will work fine only if m1() is removed from either T2 and T3.
  - None of the above.
-

---

Given :

```
//In Data.java
public class Data{
    int value;
    Data(int value){
        this.value = value;
    }
    public String toString(){ return ""+value; }
}
```

and the following code fragments:

```
public void filterData(ArrayList<Data> dataList, Predicate<Data> p){
    Iterator<Data> i = dataList.iterator();
    while(i.hasNext()){
        if(p.test(i.next())){
            i.remove();
        }
    }
}
...
Data d = new Data(1); al.add(d);
d = new Data(2); al.add(d);
d = new Data(3); al.add(d);

//INSERT METHOD CALL HERE
System.out.println(al);
```

---

Which of the following options can be inserted above so that it will print [1, 3]?

---

Please select 1 option

- filterData(al, d -> d.value%2 == 0 );
  - filterData(al, (Data x) -> x.value%2 == 0 );
  - filterData(al, (Data y) -> y.value%2 );
  - filterData(al, d -> return d.value%2 );
-

Which of the following statements are acceptable?

---

Please select 4 options

`Object o = new java.io.File("a.txt");`

(Assume that `java.io.File` is a valid class with a constructor that takes a `String`.)

---

`Boolean bool = false;`

`char ch = 10;`

---

`Thread t = new Runnable();`

(Assume that `Runnable` is a valid interface.)

---

`Runnable r = new Thread();`

(Assume that `Thread` is a class that implements `Runnable` interface)

---

Which is the earliest line in the following code after which the object created on line // 1 can be garbage collected, assuming no compiler optimizations are done?

```
public class NewClass{
    private Object o;
    void doSomething(Object s){ o = s; }

    public static void main(String args[]){
        Object obj = new Object(); // 1
        NewClass tc = new NewClass(); //2
        tc.doSomething(obj); //3
        obj = new Object(); //4
        obj = null; //5
        tc.doSomething(obj); //6
    }
}
```

---

Please select 1 option

Line 1

---

Line 2

---

Line 3

---

Line 4

---

Line 5

---

Line 6

---

Which of the following are true about the enhanced for loop?

---

Please select 3 options

- It can iterate over an array or a Collection but not a Map.

---

  - Using an enhanced for loop prevents the code from going into an infinite loop.

---

  - Using an enhanced for loop on an array may cause infinite loop.

---

  - An enhanced for loop can iterate over a Map.

---

  - You cannot find out the number of the current iteration while iterating.
-

---

Which one of these is a proper definition of a class TestClass that cannot be sub-classed?

---

Please select 1 option

- final class TestClass { }
  - abstract class TestClass { }
  - native class TestClass { }
  - static class TestClass { }
  - private class TestClass { }
-

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 (ArithmetricException 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.
-

Which of the following statements will correctly create and initialize an array of Strings to non null elements?

---

Please select 4 options

- `String[] sA = new String[1] { "aaa"};`

---

  - `String[] sA = new String[] { "aaa"};`

---

  - `String[] sA = new String[1] ; sA[0] = "aaa";`

---

  - `String[] sA = {new String( "aaa")};`

---

  - `String[] sA = { "aaa"};`
-

Given:

```
public class Swticher{  
  
    public static void main(String[] args){  
        switch(Integer.parseInt(args[1])) //1  
        {  
            case 0 :  
                boolean b = false;  
                break;  
  
            case 1 :  
                b = true; //2  
                break;  
        }  
  
        if(b) System.out.println(args[2]);  
    }  
}
```

What will the above program print if compiled and run using the following command line:

`java Swticher 1 2 3`

---

Please select 1 option

- It will print 1
  - It will print 2
  - It will print 3
  - It will not print anything.
  - It will not compile because of //1.
  - It will not compile because of //2.
  - It will not compile for some other reason.
-

Given:

```
abstract class Vehicle{ }
interface Drivable{ }
class Car extends Vehicle implements Drivable{ }
class SUV extends Car { }
```

Which of the following options will fail to compile?

---

Please select 1 option

`ArrayList<Vehicle> al1 = new ArrayList<>();  
al1.add(new SUV());`

---

`ArrayList<Drivable> al2 = new ArrayList<>();  
al2.add(new Car());`

---

`ArrayList<Drivable> al3 = new ArrayList<>();  
al3.add(new SUV());`

---

`ArrayList<SUV> al4 = new ArrayList<>();  
al4.add(new Car());`

---

`ArrayList<Vehicle> al5 = new ArrayList<>();  
al5.add(new Car());`

---

---

What happens when you try to compile and run the following program?

```
public class CastTest{  
    public static void main(String args[ ] ){  
        byte b = -128 ;  
        int i = b ;  
        b = (byte) i;  
        System.out.println(i+" "+b);  
    }  
}
```

---

Please select 1 option

- The compiler will refuse to compile it because i and b are of different types cannot be assigned to each other.
  - The program will compile and will print -128 and -128 when run .
  - The compiler will refuse to compile it because -128 is outside the legal range of values for a byte.
  - The program will compile and will print 128 and -128 when run .
  - The program will compile and will print 255 and -128 when run .
-

Which of the following can be valid declarations of an integer variable?

---

Please select 2 options

global int x = 10;

---

final int x = 10;

---

public Int x = 10;

---

Int x = 10;

---

static int x = 10;

---

What will the following code print when run?

```
class Baap {  
    public int h = 4;  
    public int getH() {  
        System.out.println("Baap " + h);  
        return h;  
    }  
}  
  
public class Beta extends Baap {  
    public int h = 44;  
    public int getH() {  
        System.out.println("Beta " + h);  
        return h;  
    }  
    public static void main(String[] args) {  
        Baap b = new Beta();  
        System.out.println(b.h + " " + b.getH());  
        Beta bb = (Beta) b;  
        System.out.println(bb.h + " " + bb.getH());  
    }  
}
```

Please select 1 option

Beta 44

4 44

Baap 44

44 44

---

Baap 44

4 44

Beta 44

44 44

---

Beta 44

4 44

Beta 44

4 44

---

Beta 44

4 44

Beta 44

44 44

Which of the following are valid declarations inside an interface independent of each other?

Please select 2 options

default void compute();

public void compute();

static void compute(){  
    System.out.println("computing...");  
}

static void compute();

default static void compute(){  
    System.out.println("computing...");  
};

Consider the following

```
public class TestClass {  
    public static void main(String[] args) {  
        TestClass tc = new TestClass();  
        tc.myMethod();  
    }  
  
    public void myMethod() {  
        yourMethod();  
    }  
  
    public void yourMethod() {  
        throw new Exception();  
    }  
}
```

What changes can be done to make the above code compile?

---

Please select 1 option

- Change declaration of main to :  
public static void main(String[] args) throws Exception
- Change declaration of myMethod to  
public void myMethod throws Exception
- Change declaration of yourMethod to  
public void yourMethod throws Exception
- Change declaration of main and yourMethod to :  
 public static void main(String[] args) throws Exception and  
public void yourMethod throws Exception
- Change declaration of all the three method to include throws Exception.

What will the following program print?

```
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.

Note: Although Wrapper classes are not explicitly mentioned in the exam objectives, we have seen some candidates get questions on this aspect of Wrapper classes.

What will be the output of the following program?

```
public class EqualTest{
    public static void main(String args[]){
        Integer i = new Integer(1) ;
        Long m = new Long(1);
        if( i.equals(m)) System.out.println("equal"); // 1
        else System.out.println("not equal");
    }
}
```

---

Please select 1 option

equal

---

not equal

---

Compile time error at //1

---

Runtime error at //1

---

None of the above.

---

Which of the following declarations is/are valid:

1. `bool b = null;`
  2. `boolean b = 1;`
  3. `boolean b = true|false;`
  - 4 `bool b = (10<11);`
  5. `boolean b = true||false;`
- 
- 

Please select 1 option

1 and 4

---

2, 3, and 5

---

2 and 3

---

3 and 5

---

5

---

Given:

```
interface Account{
    public default String getId(){
        return "0000";
    }
}

interface PremiumAccount extends Account{
    //INSERT CODE HERE
}
```

Which of the following options can be inserted in PremiumAccount independent of each other?

---

Please select 2 options

- static String getId(){  
 return "1111";  
}

---

  - String getId();

---

  - default String getId(){  
 return "1111";  
}

---

  - abstract static String getName();

---

  - static String getName();

---

  - default String getName();
-

Consider the following interface definition:

```
public interface ConstTest{  
    public int A = 1; //1  
    int B = 1;        //2  
    static int C = 1; //3  
    final int D = 1;  //4  
    public static int E = 1; //5  
    public final int F = 1; //6  
    static final int G = 1; //7  
    public static final int H = 1; //8  
}
```

Which line(s) will cause a compilation error?

Please select 1 option

1

2

3

4

5

6

7

8

None of them will cause any error.