**Q.** What will be the result of below given code?  
  
public class DoWhile {  
public static void main (String [] args) {  
int ii = 2;  
do {  
System.out.println (ii);  
} while (--ii);  
}  
}

**1)**. 1

**2)**. 2

**3)**. null

**4)**. an infinite loop

**5)**. **compilation error**

**Q.** Which declaration initializes a boolean variable?

**1)**. boolean h = 1;

**2)**. boolean k = 0;

**3)**. **boolean m = null;**

**4)**. **boolean j = (1 < 5) ;**

**Q.** What will be the result of the below code?  
  
Class X  
{  
String str = "default";  
X(String s)  
{  
str = s;  
}  
void print ()  
{  
System.out.println(str);  
}  
public static void main(String[] args)  
{  
new X("hello").print();  
}  
}

**1)**. **hello**

**2)**. default

**3)**. **The Program will print nothing**

**4)**. Compilation fails

**5)**. An exception is thrown at runtime

**Q.** Given the code fragment:  
  
System.out.println("Result: " + 2 + 3 + 5);  
System.out.println("Result: " + 2 + 3 \* 5);  
  
What will be the result?

**1)**. **Result: 235 Result: 215**

**2)**. Result: 10 Result: 30

**3)**. Result: 10 Result: 25

**4)**. **Result: 215 Result: 215**

**5)**. Compilation fails

**Q.** Given the code fragment:  
  
String valid = "true";  
if (valid)  
{  
System.out.println ("valid");  
}  
else  
{  
System.out.println ("not valid");  
}  
  
What is the result?

**1)**. Valid

**2)**. not valid

**3)**. **Compilation fails**

**4)**. An IllegalArgumentException is thrown at run time

**Q.** Which two are valid instantiations and initializations of a multi dimensional array?

**1)**. int [] [] array 2D = { { 0, 1, 2, 4} {5, 6}};

**2)**. **int [] [] array2D = new int [2] [2]; array2D[0] [0] = 1; array2D[0] [1] = 2; array2D[1] [0] = 3; array2D[1] [1] = 4;**

**3)**. int [] [] [] array3D = {{0, 1}, {2, 3}, {4, 5}};

**4)**. **int [] [] [] array3D = new int [2] [2] [2]; array3D [0] [0] = array; array3D [0] [1] = array; array3D [1] [0] = array; array3D [0] [1] = array;**

**5)**. int [] [] array2D = {0, 1};

**Q.** What will be the result?  
  
public class x  
{  
public static void main (String [] args)  
{  
String theString = "Hello World";  
System.out.println(theString.charAt(11));  
}  
}

**1)**. There is no output

**2)**. d is output

**3)**. **A StringIndexOutOfBoundsException is thrown at runtime**

**4)**. **An ArrayIndexOutOfBoundsException is thrown at runtime**

**5)**. A NullPointException is thrown at runtime

**6)**. A StringArrayIndexOutOfBoundsException is thrown at runtime

**Q.** Refer the given code:  
  
import java.io.IOException;  
public class Y  
{  
public static void main(String[] args) {  
try {  
doSomething();  
}  
catch (RuntimeException e) {  
System.out.println(e);  
}  
}  
static void doSomething()  
{  
if (Math.random() > 0.5)  
{  
throw new IOException();  
}  
throw new RuntimeException();  
}  
}  
  
Which two actions, used independently, will permit this class to compile?

**1)**. **Adding throws IOException to the main() method signature**

**2)**. **Adding throws IOException to the doSoomething() method signature**

**3)**. **Adding throws IOException to the main() method signature and to the dosomething() method**

**4)**. **Adding throws IOException to the dosomething() method signature and changing the catch argument to IOException**

**5)**. Adding throws IOException to the main() method signature and changing the catch argument to IOException

**Q.** Given the code fragment:  
  
int j=0, k =0;  
for (int i=0; i < x; i++)  
{  
do  
{  
k=0;  
while (k < z)  
{  
k++;  
System.out.print(k + " ");  
}  
System.out.println(" ");  
j++;  
} while (j < y);  
System.out.println("----");  
}  
  
What values of x, y, z will produce the following result?  
  
1 2 3 4  
1 2 3 4  
1 2 3 4  
------  
1 2 3 4  
------

**1)**. X = 4, Y = 3, Z = 2

**2)**. X = 3, Y = 2, Z = 3

**3)**. X = 2, Y = 3, Z = 3

**4)**. X = 4, Y = 2, Z = 3

**5)**. **X = 2, Y = 3, Z = 4**

**Q.** Which approach ensures that the class can be compiled and run?  
  
public class Main  
{  
public static void main(String[] args)  
{  
doSomething();  
}  
private static void doSomething()  
{  
doSomeThingElse();  
}  
private static void doSomeThingElse()  
{  
throw new Exception();  
}  
}

**1)**. **Put the throw new Exception() statement in the try block of try catch**

**2)**. Put the doSomethingElse() method in the try block of a try catch

**3)**. Put the doSomething() method in the try block of a try catch

**4)**. Put the doSomething() method and the doSomethingElse() method in the try block of a try catch

**Q.** What will make the below code compile and run?  
  
public class Simple { /\* Line 1 \*/  
public float price; /\* Line 2 \*/  
public static void main (String[] args) { /\* Line 3 \*/  
Simple price = new Simple (); /\* Line 4 \*/  
price = 4; /\* Line 5 \*/  
} /\* Line 6 \*/  
} /\* Line 7 \*/

**1)**. Change line 2 to the following: Public int price

**2)**. Change line 4 to the following: int price = new simple ();

**3)**. Change line 4 to the following: Float price = new simple ();

**4)**. **Change line 5 to the following: price.price = 4;**

**5)**. Change line 5 to the following: Price = (float) 4:

**6)**. Change line 5 to the following: Price = (Simple) 4;

**Q.** Why will the below code not compile?  
  
public class Basic {  
private static int letter;  
public static int getLetter();  
public static void Main(String[] args) {  
System.out.println(getLetter());  
}  
}

**1)**. A static field cannot be private.

**2)**. **The getLetter method has no body.**

**3)**. There is no setletter method.

**4)**. The letter field is uninitialized.

**5)**. It contains a method named Main instead of main

**Q.** Given the code fragment, what will be the result?  
  
int [] [] array2D = {{0, 1, 2}, {3, 4, 5, 6}};  
system.out.print (array2D[0].length+ "" );  
system.out.print(array2D[1].getClass(). isArray() + "");  
system.out.println (array2D[0][1]);

**1)**. 3false1

**2)**. 2true3

**3)**. 2false3

**4)**. **3true1**

**5)**. 3false3

**Q.** Refer the below code:  
  
public class MyFor3  
{  
public static void main(String[] args)  
{  
int [] xx = null;  
System.out.println(xx);  
}  
}  
  
What is the result?

**1)**. Null

**2)**. **Compilation error**

**3)**. RunTimeException

**4)**. Print nothing

**5)**. Java.lang.NullPointerException

**6)**. **0**

**Q.** What will be the result?  
  
public class DoCompare4  
{  
public static void main(String[] args)  
{  
String[] table = {"aa", "bb", "cc"};  
int ii =0;  
do  
while (ii < table.length)  
System.out.println(ii++);  
while (ii < table.length);  
}  
}

**1)**. 1 0

**2)**. **0 1 2**

**3)**. 0 1 2 3

**4)**. **Compilation fails**

**Q.** public class MyFive {  
static void main(String[] args) {  
short ii;  
short jj = 0;  
for (ii = kk;ii > 6; ii -= 1) { // line x //  
jj++;  
}  
System.out.println("jj = " + jj);  
}  
}  
  
What value should replace KK in line x to cause jj = 5 to be output?

**1)**. **11**

**2)**. 1

**3)**. 2

**4)**. -1

**5)**. 0

**Q.** Which approach ensures that the class can be compiled and run?  
  
public class Main  
{  
public static void main(String[] args)  
{  
doSomething();  
}  
private static void doSomething()  
{  
doSomeThingElse();  
}  
private static void doSomeThingElse()  
{  
throw new Exception();  
}  
}

**1)**. **Put the throw new Exception() statement in the try block of try catch**

**2)**. Put the doSomethingElse() method in the try block of a try catch

**3)**. Put the doSomething() method in the try block of a try catch

**4)**. Put the doSomething() method and the doSomethingElse() method in the try block of a try catch

**Q.** Given the code fragment:  
  
int b = 4;  
b -- ;  
System.out.println (-- b);  
System.out.println(b);  
  
What will be the result?

**1)**. **22**

**2)**. 12

**3)**. 23

**4)**. 33

**Q.**Which statement, when inserted into line "// TODO code application logic here ", is valid change?  
  
public class SampleClass  
{  
public static void main(String[] args)  
{  
AnotherSampleClass asc = new AnotherSampleClass();  
SampleClass sc = new SampleClass();  
// TODO code application logic here  
}  
}  
class AnotherSampleClass extends SampleClass  
{  
}

**1)**. asc = sc;

**2)**. **sc=asc;**

**3)**. asc = (object) sc;

**4)**. asc = sc.clone ()

**Q.**Refer the below code and choose three lines which will compile and output "right on!"?  
  
public class Speak { /\* Line 1 \*/  
public static void main(String[] args) { /\* Line 2 \*/  
Speak speakIT = new Tell(); /\* Line 3 \*/  
Tell tellIt = new Tell(); /\* Line 4 \*/  
speakIT.tellItLikeItIs(); /\* Line 5 \*/  
(Truth)speakIt.tellItLikeItIs(); /\* Line 6 \*/  
((Truth)speakIt).tellItLikeItIs(); /\* Line 7 \*/  
tellIt.tellItLikeItIs(); /\* Line 8 \*/  
(Truth)tellIt.tellItLikeItIs(); /\* Line 9 \*/  
((Truth)tellIt).tellItLikeItIs(); /\* Line 10 \*/  
}  
}  
class Tell extends Speak implements Truth {  
public void tellItLikeItIs() {  
System.out.println("Right on!");  
}  
}  
interface Truth {  
public void tellItLikeItIs()  
};

**1)**. Line 5

**2)**. Line 6

**3)**. **Line 7**

**4)**. **Line 8**

**5)**. Line 9

**6)**. **Line 10**

**Q.** class X  
{  
static void m(int i)  
{  
}  
public static void main(String[] args)  
{  
int j = 12;  
m (j);  
System.out.println(j);  
}  
}  
  
What will be the result if the above given code?

**1)**. 7

**2)**. **12**

**3)**. 19

**4)**. 8

**5)**. Compilation error

**Q.** Given the code fragment:  
  
Int [] [] array = {{0}, {0, 1}, {0, 2, 4}, {0, 3, 6, 9}, {0, 4, 8, 12, 16}};  
Systemout.printIn(array [4] [1]);  
System.out.printIn (array) [1][4]);  
int [] [] array = {{0}, {0, 1}, {0, 2, 4}, {0, 3, 6, 9}, {0, 4, 8, 12, 16}};  
System.out.println(array [4][1]);  
System.out.println(array) [1][4]);  
  
What is the result?

**1)**. 4 Null

**2)**. Null 4

**3)**. An IllegalArgumentException is thrown at run time

**4)**. **4 An ArrayIndexOutOfBoundException is thrown at run time**

**Q.** Which below line causes a compilation error?  
  
public class ScopeTest1  
{  
public static void main(String[] args)  
{  
doStuff(); // line x1  
int x1 = x2; // line x2  
int x2 = j; // line x3  
}  
static void doStuff() {  
System.out.println(j); // line x4  
}  
static int j;  
}

**1)**. line x1

**2)**. **line x2**

**3)**. line x3

**4)**. line x4

**Q.** Given a java source file:  
  
class X  
{  
X ()  
{  
}  
private void one ()  
{  
}  
}  
public class Y extends X  
{  
Y ()  
{  
}  
private void two ()  
{  
one();  
}  
public static void main (string [] args)  
{  
new Y().two ();  
}  
}  
  
What changes will make this code compile?

**1)**. adding the public modifier to the declaration of class X

**2)**. adding the protected modifier to the X() constructor

**3)**. **changing the private modifier on the declaration of the one() method to protected**

**4)**. removing the Y () constructor

**5)**. removing the private modifier from the two () method

**Q.** Given:  
  
public class TestStudent {  
public static void main(String[] args) {  
Student bob = new Student ();  
Student jian = new Student();  
bob.name = "Bob";  
bob.age = 19;  
jian = bob;  
jian.name = "Jian";  
System.out.println("Bob's Name: " + bob.name);  
}  
}  
  
What is the result when this program is executed?

**1)**. Bob's Name: Bob

**2)**. **Bob's Name: Jian**

**3)**. Nothing prints

**4)**. Bob's name

**Q.** Given the code fragment:  
  
String color = "Red";  
switch(color)  
{  
case "Red":  
System.out.println("Found Red");  
case "Blue":  
System.out.println("Found Blue");  
break;  
case "White":  
System.out.println("Found White");  
break;  
default:  
System.out.println("Found Default");  
}  
  
What is the result?

**1)**. Found Red

**2)**. **Found Red Found Blue**

**3)**. Found Red Found Blue Found White

**4)**. Found Red Found Blue Found White Found Default

**Q.** Which statement initializes a stringBuilder to a capacity of 128?

**1)**. StringBuilder sb = new String("128");

**2)**. StringBuilder sb = StringBuilder.setCapacity(128);

**3)**. StringBuilder sb = StringBuilder.getInstance(128);

**4)**. **StringBuilder sb = new StringBuilder(128);**

**Q.** What will be the result?  
  
String message1 = "Wham bam!";  
String message2 = new String("Wham bam!");  
if (message1 == message2)  
System.out.println("They match");  
if (message1.equals(message2))  
System.out.println("They really match");

**1)**. They match They really match

**2)**. **They really match**

**3)**. They match

**4)**. Nothing Prints

**5)**. They really match They really match

**Q.** public class DoCompare1 {  
public static void main(String[] args) {  
String[] table = {"aa", "bb", "cc"};  
for (String ss: table) {  
int ii = 0;  
while (ii < table.length) {  
System.out.println(ss + ", " + ii);  
ii++;  
}  
}  
  
How many times is 2 printed as a part of the output?

**1)**. Zero

**2)**. Once

**3)**. Twice

**4)**. **Thrice**

**5)**. **Compilation fails.**

**Q.** Which three declarations will compile below?  
  
Public static void main (String [] args) {  
int a, b, c = 0;  
int a, b, c;  
int g, int h, int i = 0;  
int d, e, F;  
int k, l, m; = 0;

**1)**. **int a, b, c = 0;**

**2)**. **int a, b, c;**

**3)**. **int g, int h, int i = 0;**

**4)**. **int d, e, F;**

**5)**. int k, l, m; = 0;