

What will the following code print?

```
List s1 = new ArrayList( );
s1.add("a");
s1.add("b");
s1.add("c");
s1.add("a");
System.out.println(s1.remove("a")+" "+s1.remove("x"));
```

Please select 1 option

1 0
2 -1
2 0
1 -1
true false

[Add Note](#)

Which of the following is NOT a primitive data value in Java?

Please select 2 options

<input type="checkbox"/>	"x"
<input type="checkbox"/>	'x'
<input type="checkbox"/>	10.2F
<input type="checkbox"/>	new Object()
<input type="checkbox"/>	false

[Add Note](#)

Given:

```
import java.util.*;
public class TestClass {
    public static void main(String[] args) throws Exception {
        ArrayList<Double> al = new ArrayList<>();

        //INSERT CODE HERE
    }
}
```

What can be inserted in the above code so that it can compile without any error?

Please select 2 options

<input type="checkbox"/> al.add(111);
<input type="checkbox"/> System.out.println(al.indexOf(1.0));
<input type="checkbox"/> System.out.println(al.contains("string"));
<input type="checkbox"/> Double d = al.get(al.length);

[Add Note](#)

Which of the following are valid declarations of the standard main method?

Please select 2 options

<input type="checkbox"/> static void main(String args[]) { }
<input type="checkbox"/> public static int main(String args[]) {}
<input type="checkbox"/> public static void main (String args) { }
<input type="checkbox"/> final static public void main (String[] arguments) { }
<input type="checkbox"/> public static void main (String[] args) { }

[Add Note](#)

What will the following statement print?

```
System.out.printf("I scored %d marks in the %s exam!", "90%", "Java Foundations" );
```

- Please select 1 option
- I scored 90% marks in the Java Foundations exam!
 - I scored 90 marks in the Java Foundations exam!
 - Exception will be thrown at run time.
 - It will print I scored 90 marks in the Java Foundations exam! if "90%" is changed to "90" in the argument list.

[Add Note](#)

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

Please select 2 options

<input type="checkbox"/> global int x = 10;
<input type="checkbox"/> final int x = 10;
<input type="checkbox"/> public Int x = 10;
<input type="checkbox"/> Int x = 10;
<input type="checkbox"/> static int x = 10;

[Add Note](#)

Given the following set of member declarations, which of the following is true?

```
int a; // (1)
static int a; // (2)
int f( ) { return a; } // (3)
static int f( ) { return a; } // (4)
```

Please select 2 options

- | |
|--|
| <input type="checkbox"/> Declarations (1) and (3) cannot occur in the same class definition. |
| <input type="checkbox"/> Declarations (2) and (4) cannot occur in the same class definition. |
| <input type="checkbox"/> Declarations (1) and (4) cannot occur in the same class definition. |
| <input type="checkbox"/> Declarations (2) and (3) cannot occur in the same class definition. |
| <input type="checkbox"/> Declarations (1) and (2) cannot occur in the same class definition. |

[Add Note](#)

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.

[Add Note](#)

Which of the following can be used as a constructor for the class shell given below?

```
public class TestClass
{
    // lots of code ...
}
```

Please select 2 options

- ☐ public void TestClass() {...}
- ☐ public TestClass() {...}
- ☐ public static TestClass() {...}
- ☐ public final TestClass() {...}
- ☐ public TestClass(int x) { ...}

[Add Note](#)

Which of the following methods do not follow JavaBeans naming conventions?

```
public class Bond
{
    public String ticker;
    private double Coupon;
    public String getTicker()
    {
        return ticker;
    }

    public void setTicker(String ticker)
    {
        this.ticker = ticker;
    }

    public double getCoupon()
    {
        return Coupon;
    }

    public void setCoupon(double coupon)
    {
        //do nothing
    }

    public java.util.Date getMaturity()
    {
        return new java.util.Date();
    }

    public boolean isFloater(){ return false; }

    public boolean getCallable(){ return true; }
}
```

Please select 1 option

<input type="radio"/> getTicker
<input type="radio"/> getCoupon
<input type="radio"/> setCoupon
<input type="radio"/> getMaturity
<input type="radio"/> isFloater
<input type="radio"/> getCallable
<input type="radio"/> All are valid.

[Add Note](#)

Which method declarations will enable a class to be run as a standalone program?

Please select 2 options

<input type="checkbox"/> static void main(String args[])
<input type="checkbox"/> public void static main(String args[])
<input type="checkbox"/> public static main(String[] argv)
<input type="checkbox"/> final public static void main(String [] array)
<input type="checkbox"/> public static void main(String args[])

[Add Note](#)

What will the following code print?

```
class Test
{
    public static void main(String args[])
    {
        int c = 0;
        A: for(int i = 0; i < 2; i++)
        {
            B: for(int j = 0; j < 2; j++)
            {
                C: for(int k = 0; k < 3; k++)
                {
                    c++;
                    if(k>j) break;
                }
            }
        }
        System.out.println(c);
    }
}
```

Please select 1 option

7

8

9

10

11

[Add Note](#)

Which of these statements regarding the following code are correct ?

```
public class TestClass
{
    static int a;
    int b;
    public TestClass()
    {
        int c;
        c = a;
        a++;
        b += c;
    }
    public static void main(String args[]) {    new TestClass();    }
}
```

Please select 1 option

The code will fail to compile, since the constructor is trying to access static members.

The code will fail to compile, since the constructor is trying to use static member variable a before it has been initialized.

The code will fail to compile, since the constructor is trying to use member variable b before it has been initialized.

The code will fail to compile, since the constructor is trying to use local variable c before it has been initialized.

The code will compile and run without any problems.

[Add Note](#)

In the following code what will be the output if 0 (integer value zero) is passed to loopTest()?

```
public class TestClass
{
    public void loopTest(int x)
    {
        loop: for (int i = 1; i < 5; i++)
        {
            for (int j = 1; j < 5; j++)
            {
                System.out.println(i);
                if (x == 0) { continue loop; }
                System.out.println(j);
            }
        }
    }
}
```

Please select 1 option

The program will not compile.

It will print 1 2 3 4

It will print 1 1 2 3 4

It will print 1 1 2 2 3 3 4 4

Produces no output

[Add Note](#)

What will the following program print?

```
class Test
{
    public static void main(String args[])
    {
        int c = 0;
        boolean flag = true;
        for(int i = 0; i < 3; i++)
        {
            while(flag)
            {
                c++;
                if(i>c || c>5) flag = false;
            }
            System.out.println(c);
        }
    }
}
```

Please select 1 option

3

4

5

6

7

[Add Note](#)

Consider the following class:

```
public class ArgsPrinter
{
    public static void main(String args)
    {
        for(int i=0; i<3; i++)
        {
            System.out.println(args);
        }
    }
}
```

What will be printed when the above class is run using the following command line:
java ArgsPrinter 1 2 3 4

Please select 1 option

1 2 3

ArgsPrinter 1 2

java ArgsPrinter 1 2

1 1 1

None of these.

[Add Note](#)

What can be inserted at //1 and //2 in the code below so that it will print a number between 0.0 and 1.0?
(Assume that no package has been imported in the code.)

```
//1
double d = //2
System.out.println(d);
```

Please select 2 options

java.util.Random r = new java.util.Random();
and
r.random();

Random r = new Random();
and
r.random();

java.util.Random r = new java.util.Random();
and
r.nextDouble();

Random r = new Random();
and
r.next();

java.util.Random r = new java.util.Random(100);
and
r.nextDouble();

Random r = new Random(100);
and
r.getDouble();

Add Note

Which of these is the correct format to use to create the char literal of value a?

Assume that \u0061 is the unicode value for a.

Please select 2 options

<input type="checkbox"/>	'a'
<input type="checkbox"/>	"a"
<input type="checkbox"/>	new Character(a)
<input type="checkbox"/>	\u0061
<input type="checkbox"/>	'\u0061'

[Add Note](#)

What will the following program print?

```
class Test
{
    public static void main(String args[])
    {
        int i=0, j=0;
        X1: for(i = 0; i < 3; i++)
        {
            X2: for(j = 3; j > 0; j--)
            {
                if(i < j) continue X1;
                else break X2;
            }
        }
        System.out.println(i+" "+j);
    }
}
```

Please select 1 option

0 3

0 2

3 0

3 3

2 2

[Add Note](#)

Given the following two lines of code:

```
int rate = 10;
XXX amount = 1 - rate/100*1 - rate/100;
```

What can XXX be?

See Hint

Please select 1 option

<input type="radio"/> only int or long
<input type="radio"/> only long or double
<input type="radio"/> only double
<input type="radio"/> int, long, float, or double
<input type="radio"/> long or double but not int or float.

[Add Note](#)

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) {}
}
```

[Add Note](#)

Consider the following method:

```
public static void ifTest(boolean flag)
{
    if (flag)    //1
    if (flag)    //2
    if (flag)    //3
    System.out.println("False True");
    else        //4
    System.out.println("True False");
    else        //5
    System.out.println("True True");
    else        //6
    System.out.println("False False");
}
```

Which of the following statements are correct ?

Please select 2 options

If run with an argument of 'false', it will print 'False False'

If run with an argument of 'false', it will print 'True True'

If run with an argument of 'true', it will print 'True False'

It will never print 'True True'

It will not compile.

[Add Note](#)

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.](#)

[OK](#)

[Reset](#)

[Show My Answer](#)

[Show Correct Answer](#)



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.

[Add Note](#)

Which of the following statements are true?

Please select 2 options

<input type="checkbox"/> System.out.println(1 + 2 + "3"); would print 33.
<input type="checkbox"/> System.out.println("1" + 2 + 3); would print 15.
<input type="checkbox"/> System.out.println(4 + 1.0f); would print 5.0
<input type="checkbox"/> System.out.println(5/4); would print 1.25
<input type="checkbox"/> System.out.println('a' + 1); would print b.

[Add Note](#)

What will the following code print when compiled and run?

```
import java.util.*;
public class TestClass {
    public static void main(String[] args) throws Exception {
        ArrayList<String> al = new ArrayList<String>();
        al.add("111");
        al.add("222");
        System.out.println(al.get(al.size()));
    }
}
```

Please select 1 option

<input type="radio"/> It will not compile.
<input type="radio"/> It will throw a NullPointerException at run time.
<input type="radio"/> It will throw an IndexOutOfBoundsException at run time.
<input type="radio"/> 222
<input type="radio"/> null

[Add Note](#)

You are developing a class that represents a Book. Which data type will you use for storing the ISBN number, which is an alphanumeric number, of the book?

Please select 1 option

<input type="radio"/> int
<input type="radio"/> char
<input type="radio"/> String
<input type="radio"/> None of these.

[Add Note](#)

What will the following code print when compiled and run?

```
public class TestClass{
    public static void main(String[] args){
        int[] arr = { 1, 2, 3, 4, 5, 6 };
        int counter = 0;
        for (int value : arr) {
            if (counter >= 5) {
                break;
            } else {
                continue;
            }
            if (value > 4) {
                arr[counter] = value + 1;
            }
            counter++;
        }
        System.out.println(arr[counter]);
    }
}
```

Please select 1 option

<input type="radio"/> It will not compile.
<input type="radio"/> It will throw an exception at run time.
<input type="radio"/> 5
<input type="radio"/> 6
<input type="radio"/> 7
<input type="radio"/> 8

[Add Note](#)

Identify valid for constructs:

Please select 3 options

for(;;Math.random())<0.5;){
 System.out.println("true");
}

for(;;Math.random())<0.5){
 System.out.println("true");
}

for(;;Math.random())
{
 System.out.println("true");
}

for(;;)
{
 if(Math.random()<.05) break;
}

Add Note

Which of the following code fragments are valid method declarations?

Please select 1 option

<code>void method1{ }</code>
<code>void method2() { }</code>
<code>void method3(void){ }</code>
<code>method4{ }</code>
<code>method5(void){ }</code>

[Add Note](#)

Given:
int a = 5, b = 2, c = 30;
System.out.println(a + ++b * c);

What is the result?

See Hint

Please select 1 option

65
210
180
95
Compilation failure

Add Note

What will the following program print?

```
class Test
{
    public static void main(String args[])
    {
        int k = 9, s = 5;
        switch(k)
        {
            default :
                if( k == 10) { s = s*2; }
                else
                {
                    s = s+4;
                    break;
                }
            case 7 : s = s+3;
        }
        System.out.println(s);
    }
}
```

Please select 1 option

5

9

12

It will not compile.

[Add Note](#)

Which of the following is a benefit of encapsulation?

Please select 1 option

- It allows you to add functionality by extending the class.
- It allows you to plug and play different components without changing the code.
- It allows you the change the implementation of the internal logic without changing the API.
- It allows you to couple one class with another.
- It provides control over data.

[Add Note](#)

Which of these statements concerning the charAt() method of java.lang.String class are true?

Please select 2 options

☐ The charAt() method can take a char value as an argument.

☐ The charAt() method returns a Character object.

☐ The expression char ch = "12345".charAt(3) will assign 3 to ch.

☐ The expression char ch = str.charAt(str.length()) where str is "12345", will assign 3 to ch.

☐ The index of the first character is 0.

[Add Note](#)

Which of the following are valid declarations:

Please select 3 options

<input type="checkbox"/> <code>int a = b = c = 100;</code>
<input type="checkbox"/> <code>int a, b, c; a = b = c = 100;</code>
<input type="checkbox"/> <code>int a, b, c=100;</code>
<input type="checkbox"/> <code>int a=100, b, c;</code>
<input type="checkbox"/> <code>int a= 100 = b = c;</code>

[Add Note](#)

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.

[Add Note](#)

Given the following code snippet:

```
int rate = 10;
int t = 5;
XXX amount = 1000.0;
for(int i=0; i<t; t++)
{
    amount = amount*(1 - rate/100);
}
```

What can XXX be?

Please select 1 option

int
long
only double
double or float
float

[Add Note](#)

Identify the fundamental principles of Object Oriented Programming.

Please select 2 options

Code reuse

Dependency Injection

Hiding unnecessary details

Scripting

Write once run anywhere (WORA)

[Add Note](#)

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

```
class TestClass
{
    public static void main(String args[])
    {
        boolean b = false;
        int i = 1;
        do
        {
            i++ ;
        } while (b = !b);
        System.out.println( i );
    }
}
```

Please select 1 option

The code will fail to compile because the while statement used in the code has an invalid condition expression.

It will compile but will throw an exception at runtime.

It will print 3.

It will create an infinite loop.

It will print 1.

[Add Note](#)

Consider the following two java files in /home/user directory:

```
//file A.java
package com.enthu;
class A
{
}

//file B.java
package com.foo;
class B
{
}
```

The files are compiled using the following command line:

```
javac -d /home *.java
```

Where will the class files be created?

Please select 1 option

Both A.class and B.class in /home

A.class in /home/com/enthu and B.class in /home/com/foo

Both A.class and B.class in /home/classes

No class file will be generated.

[Add Note](#)

A Java method

Please select 2 options

- cannot return multiple values.
- cannot be private.
- must take 1 or more parameters.
- must return a value.
- must exist inside a type definition.

[Add Note](#)

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

```
public class IfTest
{
    public static void main(String args[])
    {
        if (true)
        if (false)
            System.out.println("True False");
        else
            System.out.println("True True");
        }
    }
}
```

Please select 1 option

The code will fail to compile because the syntax of the if statement is not correct.

The code will fail to compile because the values in the condition bracket are invalid.

The code will compile correctly and will not display anything.

The code will compile correctly and will display 'True True'.

The code will compile correctly but will display 'True False'

[Add Note](#)

Given:

```
public class Test
{
    static int a;
    int b;

    public void incr(){
        int c = a++;
        b++;
        c++;
        System.out.println(a+" "+b+" "+c);
    }
    public static void main(String args[])
    {
        Test test = new Test();
        test.incr();
        a++;
        test = new Test();
        test.incr();
    }
}
```

What will be the output?

Please select 1 option

Compilation failure.

1 1 1
2 1 21 1 1
3 1 31 2 1
2 3 31 2 1
3 3 3[Add Note](#)

The JRE contains:

Please select 2 options

Java API
Virtual machine
debugging tools
IDE
Java Application Launcher

[Add Note](#)

Which of the following are primitive integral types in java?

Please select 4 options

<input type="checkbox"/> float
<input type="checkbox"/> nibble
<input type="checkbox"/> char
<input type="checkbox"/> byte
<input type="checkbox"/> short
<input type="checkbox"/> int
<input type="checkbox"/> natural

[Add Note](#)

Consider the following class definition:

```
public class TestClass
{
    public static void main(){ new TestClass().sayHello(); } //1
    public static void sayHello(){ System.out.println("Static Hello World"); } //2
    public void sayHello() { System.out.println("Hello World "); } //3
}
```

What will be the result of compiling and running the class?

Please select 1 option

☐

It will print 'Hello World'.

☐

It will print 'Static Hello World'.

☐

Compilation error at line 2.

☐

Compilation error at line 3.

☐

Runtime Error.

[Add Note](#)

What will the following code print?

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

[Add Note](#)

The following code snippet will not compile...

```
int i = 10;
System.out.println( i<20 ? out1() : out2() );
```

Assume that out1 and out2 have method signature: public void out1(); and public void out2();.

Please select 1 option

<input type="radio"/>	True
<input type="radio"/>	False

[Add Note](#)

What will be written to the standard output when the following program is run?

```
public class TrimTest
{
    public static void main(String args[])
    {
        String blank = " "; // one space
        String line = blank + "hello" + blank + blank;
        line.concat("world");
        String newLine = line.trim();
        System.out.println((int)(line.length() + newLine.length()));
    }
}
```

Please select 1 option

- | |
|--------------------|
| 25 |
| 24 |
| 23 |
| 22 |
| None of the above. |

[Add Note](#)

You are developing a Java rich client application that is to be installed on a lot of workstations used by the employees of your company. The users are located in various office locations across the globe. Which of the following Java technologies will be useful in delivering the application to the users over the internet?

Please select 1 option

<input type="radio"/> Java RMI
<input type="radio"/> Java Web Start
<input type="radio"/> Email
<input type="radio"/> EJB and Servlet
<input type="radio"/> Swing/AWT

[Add Note](#)

Which of the following are literals?

Please select 3 options

<input type="checkbox"/>	goto
<input type="checkbox"/>	case
<input type="checkbox"/>	break
<input type="checkbox"/>	true
<input type="checkbox"/>	false
<input type="checkbox"/>	null
<input type="checkbox"/>	Integer

[Add Note](#)

Identify the types of literals shown below.

1. 10

INVALID

2. 10L

int

double

3. 10F

char

float

4. 10D

long

String

5. '10'

6. "10"

7. 10C

OK

Reset

Show My Answer

Show Correct Answer



Given:

```
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?

Please select 1 option

☐ exception in div
☐ exception in main☐ exception in div☐ exception in main☐ exception in div☐ 0☐ Compilation failure[Add Note](#)

What will the following code print when run?

```
public class Mambo {  
  
    public static String makeItBetter(String str) {  
        return str+"!!!";  
    }  
  
    public static void main(String args[]){  
        String str = "Hi";  
        str = makeItBetter(str);  
        System.out.println(str);  
    }  
}
```

Please select 1 option

Hi!!!

Hi

Hi!!!!!!

None of these.

[Add Note](#)

Given:

```
class Acct {
    int id;
    double balance;

    public Acct(int id, double balance){
        this.id = id;
        this.balance = balance;
    }
    public void setId(int id){
        this.id = id;
    }
    public void setBalance(double balance){
        this.balance = balance;
    }
}

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

        //INSERT CCODE HERE

        System.out.println(acct.id+" "+acct.balance);
    }
}
```

What can be inserted in the above code so that it will print 10 10.0?

Please select 1 option

Acct acct = new Acct();
acct.id = 10;
acct.balance = 10.0;

Acct acct = new Acct(10, 10);

Acct acct = new Acct();
acct.setId(10);
acct.setBalance(10.0);

Acct acct = null;
acct.id = 10;
acct.balance = 10.0;

Acct acct;
acct.id = 10;
acct.balance = 10.0;

[Add Note](#)

Consider the following code:

```
class Test
{
    public static void main(String[] args)
    {
        for (int i = 0; i < args.length; i++)    System.out.print(i == 0 ? args[i] : " " + args[i]);
    }
}
```

What will be the output when it is run by giving the following command:
java Test good bye friend!

Please select 1 option

<input type="radio"/> It will print good bye friend!
<input type="radio"/> It will print good good good
<input type="radio"/> It will print goodgoodgood
<input type="radio"/> It will print good bye
<input type="radio"/> None of the above.

[Add Note](#)

What will the following code print when run?

```
List s1 = new ArrayList( ); //1
s1.add("ann");//2
if(s1.contains("ann")) //3
s1.add("ann");//4
System.out.println(s1.size()+" "+s1.indexOf("ann"));//5
```

Please select 1 option

- | | |
|--------------------------|---|
| 1 | 0 |
| 2 | 0 |
| 2 | 1 |
| 1 | 1 |
| Compilation failure | |
| an exception at run time | |

[Add Note](#)

What can be the return type of method getSwitch so that this program compiles and runs without any problems?

```
public class TestClass
{
    public static XXX getSwitch(int x)
    {
        return x - 20/x + x*x;
    }
    public static void main(String args[])
    {
        switch( getSwitch(10) )
        {
            case 1 :
            case 2 :
            case 3 :
            default : break;
        }
    }
}
```

[See Hint](#)

Please select 1 option

☐ int

☐ float

☐ long

☐ double

☐ char

☐ byte

☐ short

[Add Note](#)

Your GUI screen contains a text field for username. The String variable used to store the value entered by the user is named userName. Which of the following lines of code will you use to check whether the userName is empty or not? A value containing only spaces is also considered empty.

Please select 1 option

<code>userName.equals("")</code>
<code>userName.removeWhiteSpaces().equals("")</code>
<code>userName.trim().length == 0</code>
<code>userName.trim().equals("")</code>

[Add Note](#)

Consider the following code in file TestClass.java:

```
public class TestClass
{
    public static void main(String[] args)
    {
        System.out.println("Hello!!!");
    }
}
```

This file is compiled and run using the following command line:
java -version TestClass

What will be the output?

Please select 1 option

The Java interpreter will print the version information and then exit without running the java class.

It will print the version information followed by "Hello!!!".

It will only print "Hello!!!".

It will print an error message and exit.

[Add Note](#)