Congratulations! You passed!

Grade received 100% **To pass** 75% or higher

Go to next item

Java Proficiency Quiz 1

Total points 4

1. What is the correct syntax for initializing a variable named **dog**, of type **String**, with the value "**BARK**"? **1 / 1 point**

- ○ dog = "BARK";
- ○ int dog = "BARK";
- ◉ String dog = "BARK";
- ○ "BARK" = String dog;

Correct
Correct!

2. Look at the following line of code: **1 / 1 point**

```
1  int[] myList = { 1, 2, 3, 4, 5 };
```

What is the proper way to access the array element with the value 3?

◉
```
1  myList[2];
```

2. Look at the following line of code: **1 / 1 point**

```
1  int[] myList = { 1, 2, 3, 4, 5 };
```

What is the proper way to access the array element with the value 3?

- (●)
```
1  myList[2];
```
- ( )
```
1  myList["two"];
```
- ( )
```
1
2  myList.3;
```
- ( )
```
1
2  myList[3];
```

Correct
Correct!

3. Look at the following code: **1 / 1 point**

```
1  int i = 4;
2
```

Correct!

3. Look at the following code: **1 / 1 point**

```
int i = 4;

if (i <= 3) {
    i = 0;
} else if (i <= 6) {
    i = 1;
} else {
    i = 2;
}
```

What value would i be after this if then else statement?

- ○ null
- ○ 0
- ◉ 1
- ○ 2

Correct
Correct!

4. Look at the following function: **1 / 1 point**

```
public String reverseName( String name1, String name2 ) {
    String reversed = name2 + ", " + name1;
    return reversed;
}
```

- ○ 2

Correct
Correct!

4. Look at the following function: **1 / 1 point**

```
1  public String reverseName( String name1, String name2 ) {
2      String reversed = name2 + ", " + name1;
3      return reversed;
4  }
```

Which is the proper way to call this function?

- ○
```
1  ReverseName( "John", "Doe" )
```
- ◉
```
1  reverseName( "John", "Doe" )
```
- ○
```
1  reverseName()
```
- ○
```
1  reverseName( "John", 100 )
```

Correct
Correct!

Congratulations! You passed!

Grade received 100% **To pass** 75% or higher

Go to next item

Java Proficiency Quiz 2

Total points 4

1. Look at the following code: **1 / 1 point**

```
1  public class TestClass {
2
3      // section 1:
4      private String testName;
5
6      // section 2:
7      public TestClass( String name, int i ) {
8          this.testName = name;
9      }
10
11     // section 3:
12     public void countToThree() {
13         for (int m = 1; m <= 3; m++) {
14             System.out.println( "Count is: " + m );
15         }
16     }
17 }
```

What is defined in the denoted sections of this class?

- (●) **section 1:** member variable
  section 2: constructor
  section 3: method
- ( ) **section 1:** method
  section 2: constructor

What is defined in the denoted sections of this class?

- (●) **section 1:** member variable
  section 2: constructor
  section 3: method
- ( ) **section 1:** method
  section 2: constructor
  section 3: member variable
- ( ) **section 1:** member variable
  section 2: constructor
  section 3: class method
- ( ) **section 1:** member variable
  section 2: class method
  section 3: method

Correct
Correct! A member variable is also known as an instance variable.

2. As an established Java convention, what would it mean if the name of a variable was spelled in all uppercase? **1 / 1 point**

- ○ The variable is contains a string that has all capital letters.
- ○ Nothing. There is no such convention, and such a variable is like any other.
- ○ The variable is reserved for use by the Java environment, and you should not refer to it.
- ◉ The variable is a constant, whose value should not change.

Correct
Correct!

3. Look at the following code: **1 / 1 point**

```
1  int errorInteger = 200;
2  String comment;
3
4  switch (errorInteger) {
5      case 150:
6          comment = "Javascript error.";
7      break;
8      case 240:
9          comment = "Comment error.";
10     break;
11     case 300:
12         comment = "Function error.";
13     break;
14     case 200:
15         comment = "New error.";
16     break;
17     default:
18         comment = "No error.";
19     break;
20 }
21 System.out.println( comment );
22
```

What would be the resulting output from this code?

3. Look at the following code:

1 / 1 point

```
int errorInteger = 200;
String comment;

switch (errorInteger) {
    case 150:
        comment = "Javascript error.";
    break;
    case 240:
        comment = "Comment error.";
    break;
    case 300:
        comment = "Function error.";
    break;
    case 200:
        comment = "New error.";
    break;
    default:
        comment = "No error.";
    break;
}
System.out.println( comment );

```

What would be the resulting output from this code?

- ○ Comment error.
- ◉ New error.
- ○ Javascript error.
- ○ Function error.

Correct
Correct!

4. Look at the following class:

1 / 1 point

```
1  public class Test {
2      private String testName;
3
4      public Test( String name ) {
5          this.testName = name;
6      }
7
8      public setTestName( String name ) {
9          this.testName = name;
10     }
11 }
```

What would be the proper way to construct a Test object with member variable testName initially being "old", then later changed to "new"

- ○
```
1  Test testObj = new Test( "old" );
2  testObj.testName = "new";
```

- ○
```
1  Test testName = "old";
2  testName = "new";
```

- ○
```
1  Test testObj = new Test( "old" );
2  testObj[testName] = "new";
```

- ◉
```
1  Test testObj = new Test( "old" );
2  testObj.setTestName( "new" );
```

Correct
Correct!

Congratulations! You passed!

Grade received 100% **To pass** 75% or higher

Go to next item

Java Proficiency Quiz 3

Total points 4

1. What would be the output of this code?: 1 / 1 point

```
1  int age;
2  age = age + 7;
3  System.out.println( "My age is: " + age );
```

- (●) This will not compile because age is not initialized with a value.
- ( ) My age is : null
- ( ) This will not compile because you cannot add a string and integer value together.
- ( ) My age is: 7

Correct
Correct!

2. How many times will "Hello" be printed out?: 1 / 1 point

```
1  int i = 1;
2  while (i <= 3) {
3      System.out.println( "Hello" );
4      if ( i < 2 ) {
5          System.out.println( "Hello" );
6      }
```

- My age is : null
- This will not compile because you cannot add a string and integer value together.
- My age is: 7

Correct
Correct!

2. How many times will "Hello" be printed out?: **1 / 1 point**

```
int i = 1;
while (i <= 3) {
    System.out.println( "Hello" );
    if ( i < 2 ) {
        System.out.println( "Hello" );
    }
    i++;
}
---------
```

- (•) 4
- ( ) 5
- ( ) 2
- ( ) 0

Correct
Correct!

3. Look at the following code:

1 / 1 point

```
class Person {
    public void talk() {
        System.out.println( "Hello" );
    }
}

class Baby extends Person {
    public void talk() {
        System.out.println( "Goo goo" );
    }
}

class Boy extends Person {
    public void hi() {
        System.out.println( "hi" );
    }
}

public class Test{
    public static void main( String args[] ) {
        Person p1 = new Person();
        p1.talk();

        Baby b1 = new Baby();
        b1.talk();

        Boy b2 = new Boy();
        b2.talk();
        b2.hi();
    }
}
```

What would be the output of running the code above?

- (●) Hello

  Goo goo

  Hello

  hi

- ( ) Hello

What would be the output of running the code above?

- (●) Hello
  Goo goo
  Hello
  hi
- ( ) Hello
  Goo goo
  hi
  hi
- ( ) There will be no output since these are just class definitions.
- ( ) Error, because the Boy class does not have the method talk.

Correct
Correct!

4. Look at the following lines of code: **1 / 1 point**

```
1  interface MyInterface {
2      public void method1();
3      public void method2();
4  }
5
6  class MyClass implements MyInterface {
7      public void method1() {
8          System.out.println( "method 1" );
9      }
```

4. Look at the following lines of code:

1 / 1 point

```
interface MyInterface {
    public void method1();
    public void method2();
}

class MyClass implements MyInterface {
    public void method1() {
        System.out.println( "method 1" );
    }
}

public class Test {
    public static void main( String args[] ) {
        MyClass c1 = new MyClass();
        c1.method1();
        c1.method2();
    }
}
```

What would be the output of running the code above?

- ○ method 1

  method 2

- ◉ Error because there is no method2 defined in MyClass. Because this class implements MyInterface, it must define all methods that are in the interface.

- ○ method 1

  method 1

- ○ There will be no output since these are just classes definitions.

Correct

Correct!