Questions on Constructors, Objects, and Instantiation.

Elmer, Ethan, Lakshay and Tommy

Question 1

What is the relationship between classes and objects?

Question 2

What is a constructor? What is the purpose of a constructor in a class?

Question 3

For this problem, you should write a very simple but complete class. The class represents a counter that counts 0, 1, 2, 3, 4,.... The name of the class should be Counter. It has one private instance variable representing the value of the counter. It has two instance methods: increment() adds one to the counter value, and getValue() returns the current counter value. Write a complete definition for the class, Counter.

Question 4

This problem uses the Counter class from Question 3. The following program segment is meant to simulate tossing a coin 100 times. It should use two Counter objects, headCount and tailCount, to count the number of heads and the number of tails. Fill in the blanks so that it will do so.

```
Counter headCount, tailCount;
tailCount = new Counter();
headCount = new Counter();
for ( int flip = 0; flip <100; flip++ ) {
    if (Math.random() <0.5) { // There’s a 50/50 chance that this is true.
        Your Code Here ; // Count a ”head”.
    }
    else {
        Your Code Here ; // Count a ”tail”.
    }
}
System.out.println(”There were ” + Your Code Here + ” heads.”);
System.out.println(”There were ” + Your Code Here + ” tails.”);
```

Question 5

For this following question, we are going to instantiate a new monkey. And your task is to see what is the out put. Assume the Monkey class is already implemented. It takes a size and how many bananas it eats.

Here it is:

```
public static void main(String[] args) {
    Monkey champ = new Monkey(10, 200);
    Monkey biggerchamp = champ;
    biggerchamp.size = 40;
    System.out.println(champ.size);
}
```

What is the output? Does this change the size of the champ monkey?

Questions on Constructors, Objects, and Instantiation. (Solutions)

Elmer, Ethan, Lakshay and Tommy

Question 1

What is the relationship between classes and objects?

A class defines the structure and behaviors of all entities of a given type, while an object is one particular "instance" of that type of entity. For example, if Car is a class, then Ferrari would be an object of Car.

Question 2

What is a constructor? What is the purpose of a constructor in a class?

A constructor is a special kind of method in a class. It has the same name as the name of the class, and it has no return type, not even void. A constructor is called with the new operator in order to create a new object. Its main purpose is to initialize the newly created object, but it can actually do anything the programmer tells it to do.

Question 3

For this problem, you should write a very simple but complete class. The class represents a counter that counts 0, 1, 2, 3, 4,.... The name of the class should be Counter. It has one private instance variable representing the value of the counter. It has two instance methods: increment() adds one to the counter value, and getValue() returns the current counter value. Write a complete definition for the class, Counter.

One example:

```
public class Counter {
    private int value = 0; // Current value of the counter.
    public void increment() {
        // add one to the value of the counter
        value += 1;
    }
    public int getValue() {
        // get the current value of the counter
        return value;
    }
}
```

Question 4

This problem uses the Counter class from Question 3. The following program segment is meant to simulate tossing a coin 100 times. It should use two Counter objects, headCount and tailCount, to count the number of heads and the number of tails. Fill in the blanks so that it will do so.

```
Counter headCount, tailCount;
tailCount = new Counter();
headCount = new Counter();
for ( int flip = 0; flip <100; flip++ ) {
    if (Math.random() <0.5) { // There’s a 50/50 chance that this is true.
        headCount.increment(); // Count a ”head”.
    }
    else {
        tailCount.increment(); // Count a ”tail”.
    }
}
System.out.println(”There were ” + headCount.getValue() + ” heads.”);
System.out.println(”There were ” + tailCount.getValue() + ” tails.”);
```

Question 5

For this following question, we are going to instantiate a new monkey. And your task is to see what is the out put. Assume the Monkey class is already implemented. It takes a size and how many bananas it eats.

Here it is:

```
public static void main(String[] args) {
    Monkey champ = new Monkey(10, 200);
    Monkey biggerchamp = champ;
    biggerchamp.size = 40;
    System.out.println(champ.size);
}
```

What is the output? Does this change the size of the champ monkey?

10. No.

Java Compilation Process Questions

Sreesha Venkat and Nicki Peternel

February 15, 2015

1 Question 1

Fill out the blanks for the Java Compilation Process.

Java __________	.____ file
Java ___________	(e.g. javac)
Java ________ __________	.______ file
Java __________ ________	(Java Interpreter)

2 Question 2

public static void main(String[] args)

Given the line of code above, explain what each of the following keywords means in the context of Java.

- public:
- static:
- void:
- main:

3 Question 3

In Java, is it absolutely necessary for the file name to be the same as the class name? If so, explain why.

4 Question 4

If a programmer has the source code for a Java program already written, and they wish to run that program, they will need both a **compiler** and **interpreter**.
Briefly explain what a Java compiler and a Java interpreter will do:

5 Question 5

Explain three different advantages of Java using bytecode rather than native code.

Java Compilation Process Questions

Sreesha Venkat and Nicki Peternel

February 17, 2015

1 Question 1

Fill out the blanks for the Java Compilation Process.

Java **Source Code**	**.java** file
Java **Compiler**	(e.g. javac)
Java **Byte Code**	**.class** file
Java **Virtual Machine**	(Java Interpreter)

2 Question 2

public static void main(String[] args)

Given the line of code above, explain what each of the following keywords means in the context of Java.

- public: allows universal access
- static: method will be called using a class name
- void: does not return anything
- main: the name of the method

3 Question 3

In Java, is it absolutely necessary for the file name to be the same as the class name? If so, explain why.
Yes because the filename should always reflect the public class defined within that file. Otherwise, you will get a compiler error.

4 Question 4

If a programmer has the source code for a Java program already written, and they wish to run that program, they will need both a **compiler** and **interpreter**.
Briefly explain what a Java compiler and a Java interpreter will do:
The Java compiler will run through the code and check for errors. All at once, it will produce a machine language program that can be executed. The Interpreter will run through the given code one line at a time and executes immediately.

5 Question 5

Explain three different advantages of Java using bytecode rather than native code.
Portability, Security, Size

1 Question 1

Go through the program and trace the execution. What is the output after the program runs?

```
public class Autobots extends Transformers {
    public drive() {
        optimus(2001);
    }
    private void optimus(int y) {
        int x = y / 1000;
        int z = (x + y);
        x = bumblebee(z, y);
        System.out.println(”bumblebee: s = ” + x + ”, y = ” + y + ”, z = ” + z);
    }
    private int bumblebee(int x, int y) {
        int z = jazz(x + y, y);
        y = y / z;
        System.out.println(”bumblebee: x = ” + x + ”, y = ” + y + ”, z = ” + z);
        return z;
    }
    private int jazz(int x, int y) {
        y = x (x System.out.println(”jazz: x = ” + x + ”, y = ” + y);
        return y;
    }
}
```

2 Question 2

```
public static boolean method1(int[] array, int target) {
    for (int i=0; i¡array.length;i++) {
        if (array[i] == target) return true;
        else if (array[i] ¿ target) return false;
    }
    return false;
}
public static boolean method2(int[] array, int target) {
    int low = 0;
    int high = array.length - 1;
    while (low ¡= high) {
        int mid = (low+ high) / 2;
        if (array[mid] == target) return true;
        else if (array[mid] ¡ target) low = mid + 1;
        else high = mid - 1;
    }
    return false;
}
int[] a = 2, 5, 11, 14, 15, 27, 31;
```

What does method1(a, 5) return?
What does method2(a, 20) return?
Explain what each of these methods do.

3 Question 3

What is the value of x[0][0] after the following lines are executed?

```
public class Chocolate {
    public static void main(String[] args) {
        int[] y = {2, 5, 9};
        int[] [] x = {{5, 12, 10}, {8, 3, 2}, {7, 2, 1}};
        x[0] = y;
        y[1] = x[1][0];
        y[0] = x[0][1];
        x[0][0] = y[1];
        x[1] = y; int[][] z = new int[3][];
        z[0] = x[0];
        z[1] = x[1];
        z[0][0] = -z[0][0];
        z[0][0] = x[1][1];
    }
}
```

1 Question 1

Go through the program and trace the execution. What is the output after the program runs?

```
public class Autobots extends Transformers {
    public drive() {
        optimus(2001);
    }
    private void optimus(int y) {
        int x = y / 1000;
        int z = (x + y);
        x = bumblebee(z, y);
        System.out.println(”bumblebee: s = ” + x + ”, y = ” + y + ”, z = ” + z);
    }
    private int bumblebee(int x, int y) {
        int z = jazz(x + y, y);
        y = y / z;
        System.out.println(”bumblebee: x = ” + x + ”, y = ” + y + ”, z = ” + z);
        return z;
    }
    private int jazz(int x, int y) {
        y = x (x System.out.println(”jazz: x = ” + x + ”, y = ” + y);
        return y;
    }
}
```

```
jazz: x = 4004, y = 1001
bumblebee: x = 2003, y = 1, z = 1001
optimus: x = 1001, y = 2001, z = 2003
```

2 Question 2

```
public static boolean method1(int[] array, int target) {
    for (int i=0; i¡array.length;i++) {
        if (array[i] == target) return true;
        else if (array[i] ¿ target) return false;
    }
    return false;
}
public static boolean method2(int[] array, int target) {
    int low = 0;
    int high = array.length - 1;
    while (low ¡= high) {
        int mid = (low+ high) / 2;
        if (array[mid] == target) return true;
        else if (array[mid] ¡ target) low = mid + 1;
        else high = mid - 1;
    }
    return false;
}
int[] a = 2, 5, 11, 14, 15, 27, 31;
```

What does method1(a, 5) return?
True
What does method2(a, 20) return?
False
Explain what each of these methods do.
check that the variable 'target' is in the array passed in

3 Question 3

What is the value of x[0][0] after the following lines are executed?

```
public class Chocolate {
    public static void main(String[] args) {
        int[] y = {2, 5, 9};
        int[] [] x = {{5, 12, 10}, {8, 3, 2}, {7, 2, 1}};
        x[0] = y;
        y[1] = x[1][0];
        y[0] = x[0][1];
        x[0][0] = y[1];
        x[1] = y; int[][] z = new int[3][];
        z[0] = x[0];
        z[1] = x[1];
        z[0][0] = -z[0][0];
        z[0][0] = x[1][1];
    }
}
```

8

Q/A

Jacqueline Garcia
Emily Zhang
Anokhi Kastia
Nikita Rau

Feb 02, 2015

1 The Keyword ”static”

1. What does the ”static” keyword mean?

   (a) It is associated with objects of class.

   (b) It is used to show that a method cannot be changed.

   (c) It is shared by all instances of a class.

   (d) It is a method that can be called by the class itself, and does not have to be instantiated.

   (e) a, c, and d

   (f) All of the above.

2. What should the static methods be used?

   (a) As sparingly as possible.

   (b) When you don’t want the method to stay static.

   (c) When you have a variable or value that represents some state of class (all instances of the class would have the same value for that variable).

   (d) Whenever you need to call a final class.

3. Can you use the ”static” key word in an abstract method?

   (a) Yes, but not including the modifiers: final, static, native, synchronized, or private.

   (b) No, and the following modifiers cannot be used either: final, static, native, synchronized, or private.

(c) Yes, including the modifiers (final, static, native, synchronized, or private).

(d) No, but you may use the modifiers mentioned above.

4. Can you use ".this" in a static method?

   (a) Yes.

   (b) No.

5. Explain why you cannot have a static method in a "non-static" inner class.

Special traits of java

Julie Ian Alejandro Brandon

February 10, 2015

1 Main Focus

The Java Language (special traits, why you would use it)

1.1 Question

- What the differences of using static and non-static?
- What is one difference between an Object Oriented Programing Language like Java and a Multi-paradigm Programming Language like Python?
- What is the relationship between the Java Programming Language, Bytecode, and the Java Virtual Machine?
- What is it about this ternary relationship that makes it very useful?
- Why in particular is Java?s memory management such a huge achievement of this language in particular?

1.2 Answer

- A static method belongs to the class itself and a non-static (aka instance) method belongs to each object that is generated from that class. If your method does something that doesn't depend on the individual characteristics of its class, make it static (it will make the program's footprint smaller).
- In a language like Java, all objects must have a class to which they are children, and in Python, objects may live in thier frames.
- Together, these three things make up what is known as the Java platform. Code written in Java is compiled into Bytecode. Bytecode (which in essence is low-level code that is much more abstract than regular Java code) then acts as an instruction

set for the Java Virtual Machine. Finally, the Java Virtual Machine (quite literally a virtual computer) takes in all the bytecode and translates it into machine code, but not before proof checking the code first (hence the strict compilation regulations when programming in Java).

- Together, these three things make it possible for Java programs to essentially run (at least theoretically) on any machine that has the Java Runtime Environment, thus making the porting of applications from one operating system to the next much less strenuous.

- Java uses a technique called Garbage Collection. Programmers do not have to deal with the problem of memory allocation. This technique works by tracking live objects, and regarding everything else as garbage. When an object is no longer used, the garbage collector reclaims the memory and reuses it for future memory allocation. Once an object is no longer referenced and therefore is not reachable by code, the garbage collector removes it and reclaims the unused memory.