Name: Andrew Janusz ko

Stage 4 Post

Fill in The Blank From the Vocabulary List

1. Constructors are the things that the system manages at run time.

2. When we create a name for a variable, method, or class, we should pay attention to Naming Convention so our code is more readable.

3. If we want to create an object we must use the new statement.

4. A Runnable class has a main method.

Be the Machine

Suppose your have written a class named MyClass whose toString method looks like this:

```
public String toString()
{
      return "This is one fancy object";
}
```

Output
"I really like This is one fancy object"

What would be the output from this code snipper?

```
MyClass x = new MyClass();
System.out.println("I really like " + x.toString());
```

Short Answer

5. If you don't define toString, what is the format of the output you will get if you pass an object to System.out.println()?

I really like This is one fancy object

4 yy x

6. Suppose you wanted to make a name based on the phrase "my silly name" in Java. What would be the correct capitalization if it was for a
   a. method mySillyName
   b. variable mySillyName
   c. class MySillyName
   d. constant MY_SILLY_NAME

7. Give the three definitions of the word "class."

- A type we are declaring
- The set of objects of a particular type
- The java code defining the class

8. We know a method is a constructor if what two things are true?

if it does not have a return type and its name is the name of the class

9. Suppose you are building a class that has two instance variables that are named henry and sally. Both hold real numbers. Show the code for a constructor that has parameters that pass in values that you store into those instance variables.

```
Public Class Yeet {
    Public double henry;
    Pubic double sally;

    Public Yeet (int h, int s){
    henry = h;
    Sally = s;
    }

}
```

$q^x = 2$

10. How do you know that a variable is an instance variable?

When it is private so that each object gets a copy of it

11. Consider this class:

```
public class ExampleClass
{
    private int myVar;

    public void changeItTo(int newValue)
    {
        myVar = newValue;
    }
}
```

myVar[~~new value~~][42]

x[

y[

Draw the memory diagram for this code:

```
ExampleClass x;
x = new ExampleClass();
x.changeItTo(42);
ExampleClass y;
y = new ExampleClass();
y.changeItTo(55);
```

x[42]

y[55]

myVar[newValue]