Module 6 quiz on Object Oriented Programing concepts

LATEST SUBMISSION GRADE

100%

1. **1 / 1 point**

An object is

- (•) The grouping together of data and behavior to create a single entity
- ( ) The blue print for creating classes
- ( ) A way to hide implementation details from the user
- ( ) A sequence of characters

✓ **Correct**

2. **1 / 1 point**

A typicial object oriented program

- ( ) uses objects to model the behavior of the ints, chars and boolean variables used in the program.
- ( ) uses methods and primitive data types to perform most of its useful behavior.
- (•) uses objects to perform most of its useful behavior.
- ( ) must consist of at least four classes.

✓ **Correct**

3. **1 / 1 point**

An example of abstraction would be

- ( ) supplying a Quick Start guide with a digital camera

supplying a Quick Start guide with a digital camera.

supplying the technical drawings with a digital camera.

only selling digital cameras to experienced users.

supplying batteries with a digital camera.

Correct

4. **1 / 1 point**

An example of an instance of the City class would be

the city of Philadelphia.

the variables Name, Latitude, Longitude, Country and Population.

the plans for a city.

the methods to update the city's population and to calculate the distance to other cities.

Correct

5. A class file is *(select all that apply)* **1 / 1 point**

a file containing a single program or module.

Correct

You have already written several class files of this type in previous modules of this MOOC.

a template or blueprint for an object.

Correct

This is the definition of a class file in the context of Object Oriented Programming.

- [ ] a collection of objects.
- [ ] a collection of keywords.

6. A String object is *(select all that apply)* **1 / 1 point**

- [x] a sequence of characters.

✓ **Correct**
This describes the structure or state of a String.

- [x] similar to primitive data types in some respects.

✓ **Correct**
There are many similarities between Strings and primitive data types, not only because all objects have some things in common with primitive data types but also because String objects have a few shortcuts, like how they are declared. One reason is because variables of String type are used so often, the Java language included a few conveniences.

- [x] a reference to a memory location where data is stored.

✓ **Correct**
Because a String is an object, the state is not stored in the variable itself, but is stored elsewhere in memory. One way to remember this concept is to visualize a *stack* in memory where variables are stored. All variables are expected to be a given size. Strings are of varying sizes and so, are stored elsewhere.

- [ ] a primitive data type.

7. Select all of the Java statements that would compile *(would not cause an error)*. **1 / 1 point**

- [ ]

```
1  out.println("Hello kids" - "kids");
```

```
1  String str1 = "hi";
2  String str2 = "HI";
3  out.println(str1 == str2);
```

Correct

Although comparing two Strings with the "==" operator is not ideal (in most cases we really want to determine if the two Strings have the same sequence of characters), it is syntactically permissible and will not cause a compile error. Beware!

```
1  String str1 = "Good programming";
2  out.println(str1.concat(18.9));
```

```
1  out.println("Hello" + " programmers!");
```

Correct

```
1  String str1 = "hi";
2  String str2 = "HI";
3  out.println(str1.equals(str2));
```

Correct

This is the correct way to determine if two Strings have the same sequence of characters.

8. Given the following objects **1 / 1 point**

```
1  String str1 = "held";
2  String str2;
```

How could you create the string **herald**?

```
1   str2 = str1.substring(0,2);
2   str2 = str2 + "ra";
3   str2 = str2.concat(str1.substring(2,4));
```

```
1   str1 = str1.substring(0,2) + "ra";
2   str1 = str1 + str1.substring(2,4);
```

```
1   str2 = str1 + "ra";
```

```
1   str1.substring(0,2);
2   str2 + "ra";
3   str2.concat(str1.substring(2,4));
```

Correct

Here we *insert* "ra" into the middle of the string in three steps.