Module 6 quiz on arrays and parameters

LATEST SUBMISSION GRADE

100%

1. Which code segment illustrates a correct way to declare an array of 10 objects. Assume the class Beverage has been created. **1 / 1 point**

```
Beverage[10] guestList;
```

```
new guestlist[10];
```

```
Beverage guestlist = array[10];
```

```
Beverage[] guestList = new Beverage[10];
```

Correct

Correct! This is quite similar to declaring an array of primitive data types, but remember to use the keyword new.

2. Consider the following **incorrect** code segment. Assume the Pet class has been defined and contains the method setName(String). **1 / 1 point**

```
Pet customer[] = new Pet[5];
customer[3].setName("Spot");
```

We learned that this code is incorrect because...*select the best explanation*

- The proper syntax for a set method called on an object is

```
setName(customer[3], "Spot");
```

- all five of the names must be set at one time.
- although the array of Pet references has been created, none of them point to Pet objects yet. Each object must be instantiated individually.
- objects stored in an array can not be accessed individually. They can only be accessed as a group.

Correct

Correct! A for loop is often used to instantiate each object of an array.

3. Select all of the code segments that properly create an array of 3 objects and instantiate them. Assume the class Lesson has been created. **1 / 1 point**

- [x]
```
Lesson [] week1 = new Lesson[3];
for (int i = 0; i<week1.length; i++){
    week1[i] = new Lesson();
}
```

Correct

Correct. The first line creates the array of references to the objects. The for loop then instantiates each Lesson object using the default constructor.

- [x]
```
Lesson [] week1 = {new Lesson(), new Lesson(), new Lesson()};
```

Correct

Correct. Here we create the array of references and instantiate each new object all in one line of code.

- [ ]
```
Lesson [] week1 = new Lesson[0], new Lesson[1], new Lesson [2];
```

- [ ]
```
Lesson[3] = week1[0], week1[1], week1[2];
```

4. **1 / 1 point**

When an object is passed as a parameter, its state is changed within the method that it was passed to because...

- ( ) the value of the object (its state) was passed.
- ( ) the object was updated when the return statement was executed.
- ( ) a copy of the object was passed.
- (•) the reference to the object was passed.

Correct

Correct. When passing an object as a parameter, it is passed by reference, meaning the information about the memory location where the actual object data is stored is passed to the method. Thus a change to the object in the method is a change to the actual object state.

5. 1 / 1 point

Consider the code

```
1   Pet sparky = new Pet();
2   Pet fido = sparky;
```

Line 2 of this code...

- creates a new object named sparky with the same state data found in fido.
- (•) sets the object fido to reference or "point to" the object sparky.
- creates a new object named fido with the same state data found in sparky.
- compares the object fido do the object sparky and returns true or false.

✓ **Correct**

Correct. We are in essence taking the memory location reference of sparky and copying it into the variable fido.

6. Select all of the scenarios where using a static class constant would be appropriate. 1 / 1 point

- [x] To declare a constant that is used in different methods of the class and whose value never changes.

✓ **Correct**

Correct. A good example of this was the RTNumber which was the routing number for our bank.

- [x] To declare a constant that is somehow associated with the class and may be accessed by client programs.

✓ **Correct**

Correct. A good example of this is the constant PI which is found in the Math class. As long as it is declared public it can be accessed by client programs.

- [ ] To declare a variable that is used in different methods of the class. This helps to avoid having to pass and return it as a parameter.
- [ ] To share the state of that variable across many different objects of the class, like a count variable to keep track of how many objects have been created.