New York University

CS-GY 9053 Intro To Java

MIDTERM

Name:Adith Thaniserikaran

Netid-at6115

-

Part 1-Written Questions

1. **What's the difference between a primitive type passed as a parameter into a method versus an array passed as a parameter into a method?**

Ans) When the primitive type passed as a parameter to the method it is passed by value ie the copy of the primitive type is passed to the method and any changes made in the method wont reflect on the original data.Whereas when an array is passed by reference not a copy so changes made to the array within the method will modify the original array.

2. **Back in lecture 2, the FullFeaturedCircle class has a field numObjects with an initial value of 0. In the constructor, it gets incremented. Why does the value persist every time a new object is created? That is, when you create a 4th FullFeaturedCircle object, why does the numObjects field have a value of "3" when the code enters the constructor?**

   As a reminder, here's an excerpt from the code:

```
public class FullFeaturedCircle {

	private static int numObjects = 0;
	private int circleID;
	private double radius = 1.0;

	public FullFeaturedCircle() {
		this.circleID = FullFeaturedCircle.numObjects;
		FullFeaturedCircle.numObjects++;
	}

	public FullFeaturedCircle(double radius) {
		this();
		this.radius = radius;

	}
}
```

Ans) This is because the variable numObjects is a static variable and only one single copy of this variable is shared between all the objects of the above class.Hence everytime a new object is created the variable numObjects gets incremented this is why the value persists and numObjects has a value of 3 on entering the constructor during the creation of the 4th object.

3. Explain what the this keyword means and where it is used (give at least two examples).

Ans)this is used to refer the current object in a method or in a constructor especially when the method parameters and instance variables have the same name this keywords helps us in differentiating between the two.

Examples:

1)
```
Employee(String name)
{
   this.name=name;

}
```

2)It can even be used to refer to another contructor in the same class

```
Employee()
{
String g=”Adi”;
   this(g);

}

 Employee(String name)
{
   this.name=name;

}
```

4. **What is the difference between overloading and overriding a method?**

Ans) Overloading a method is when multiple methods with the same name have different function signature ie argument list.So compiler will determine which method is to be invoked based on the give parameters.

In the case of overriding of a method the function signature and no of parameters and the name is exactly the same for example the subclass say has the same implementation of the method that exists in the super class with the exact same function signature this method can override the method of the superclass.

5. **How would you use the super keyword? Give two examples.**

Ans) The super keyword essentially is used to refer to the parent class of the current class ie superclass.It can be used to refer to the parent class's constructor or access the methods and fields which were overridden by the subclass.
Examples:

```
Employee(String name ,double salary)
{

super(name);
this.salary=salary;

}
```

2)

```
public int getavg(int a,int b)
{
   int g=super.getavg(a,b);
   return g;

}
```

6. **Let’s say we have two Strings, declared and created this way:**

   String s1 = new String(“Hello”);
   String s2 = new String(“Hello”);

   Why does s1 == s2 return false but s1.equals(s2) return true?

Ans)s1==s2 returns false because they both aren’t the same object and don’t have the he same memory addresses the equality operator checks if they are the same object. Whereas s1.equals(s2) returns true since it checks the contents of both the string objects ie “Hello” which are equal to each other.

7. **What is the difference between a checked and an unchecked exception? If you create your own Exception subclass as a subclass of Exception, it is Checked. How would you instead make it an Unchecked Exception?**

Ans) The difference between a checked and unchecked exception is the following. Checked exceptions are those exceptions which the compiler catches during compile time itself it forces you to either use try catch block to handle these exceptions or use the throws keyword.Until we don’t handle these exceptions our program wouldn’t compile itself. Unchecked exceptions are exceptions that the compiler is unable to check during compile time.They occur during the runtime such as dividing a no by 0 etc due to programming errors ie Arithmetic exception.
If we wish to make our own custom exception class as unchecked we just need to extend the RuntimeException class or any of its subclasses ie make them the super class of our custom exception class.

8. **Ontologically speaking, when creating a class, why would you make it a subclass of another class, and why would you have the class implement an interface? What is the relationship of a class to its superclass versus what is the relationship of a class to its interface?**

Ans) If I wish to extend certain properties of a particular class or extend the entire class itself with more specific usecase I can make my own current class the subclass of the class whose properties I wish to inherit.I don't need to implement the same properties in my subclass since its inherited from the parent class.The relationship between a sublass and superclass is that the subclass inherits properties of the superclass.

Interface on the other hand is implemented by a class and it forces the class to specify a set of methods if it is implementing the interface.A Class can implement multiple interfaces on the condition that it follows the rules of specifying the required methods.The relationship between a class and an interface is that the class must provide the methods defined in the interface.

9. **Take that object from an array declared as Object[] objs. The result of (new Object()).toString() is something like java.lang.Object@15db9742 - the object type followed by the @ sign and a unique identifier. In the objs array, let's say we execute the following:**

```
objs[5] = new Integer(20);
Object myObj = objs[5];
```

What is the output of myObj.toString() ? Explain why.

Ans) The output of myObj.toString() will be "20" since firstly we assign the element of the objs as a reference to the Integer object.Then when we call myObj.toString() we are calling the toString method of the Integer object which will convert the integer 20 into a string "20" and return.Hence the output is "20".

10. What is an abstract class?

Ans) Abstract classes are those classes that cannot be instantiated ie the objects cannot be created.The abstract keyword is used to make a class abstract.It can be used as a base class for various other classes.It can have abstract methods but each and every abstract method must be implemented in the subclasses.It can provide common methods implementation to all its sub classes.