CS 116
Week 8: Outline

Reading:
1. Dale, Chapter 11
2. Dale, Lab 11

Objectives:
1. Mid-term exam
1. Mid-term Exam
1. No lab this week
Question 1:

Part A (5 points)
List all the primitive integer types in Java.

Part B (5 points)
Why is there more than one primitive integer type in Java?

Question 2 (15 points):
Give concise definitions of the following terms:

a) scope

b) exception

c) abstract class

d) OOP
Question 3 (10 points):
Briefly describe the two main purposes of classes in Java.

Question 4 (10 points):
Discuss shortly the relationship between superclasses and subclasses in Java. Give a small example.

Question 5 (10 points):
Suppose you have a class named Account that represents a bank account and whose constructor accepts a parameter of type double that represents the initial deposit into the account. Write the statement(s) that would declare and create an Account object named myAccount with an initial deposit of $500.00.

Question 6 (10 points):
Suppose the simple class Point2D, which is use to create objects which are ordered pairs of real numbers, is defined as follows:
public class Point2D
{
    public double x, y;
}
Write a subclass of Point2D named Point3D which inherits Point2D's instance variables and adds an instance variable z (and thus represents ordered triples of real numbers).
Question 7 (15 points):
What output will the following program generate?

```java
public class ArrayStuff {
    private static void changeThings (int[] array) {
        array[3] = 100;
        array = new int[10];
        array[8] = -20;
    }

    public static void main (String[] args) {
        int[] x = new int[5];
        x[2] = 3;
        changeThings(x);
        for (int index = 0; index < x.length; index++) {
            System.out.println("x[" + index + "] = " + x[index]);
        }
    }
}
```
Question 8 (20 points):
Write a complete class called PairOfDice that can be used to create objects which represent pairs of dice. Your class should provide instance variables which can only be accessed through methods. Your class should provide a roll() method which will roll the dice, getDie1() and getDie2() methods which will return the value of each die individually, and a getTotal() method which will return the total of the two dice. Your class should also provide a constructor that will use the class's roll() method to create a pair of dice with random initial values.
Question 1: Part A:

byte
short
int
long

Question 1: Part B:

Each type requires a different amount of memory storage. The purpose of this is to use the appropriate amount of memory space for the given range of integers that will be used in the program. We do not want to use an excessive amount of memory if not needed.

Question 2:

a) scope - A characteristic of an identifier that determines where the identifier can be used. Most identifiers in the Java(TM) programming environment have either class or local scope. Instance and class variables and methods have class scope; they can be used outside the class and its subclasses only by prefixing them with an instance of the class or (for class variables and methods) with the class name. All other variables are declared within methods and have local scope; they can be used only within the enclosing block.

b) exception - An event during program execution that prevents the program from continuing normally; generally, an error. The Java(TM) programming language supports exceptions with the try, catch, and throw keywords.

c) abstract class - a class that is not used to construct objects, but only as a basis for making subclasses. An abstract class exists only to express the common properties of all its subclasses.

d) OOP - Object Oriented Programming - programming that focuses on using software objects that model real and/or abstract objects (like a pair of dice), instead of just performing tasks.

Question 3:

Answer: Classes in Java have two main purposes:
- To hold collections of subroutines and variables (static subroutines and variables). These include programs and many classes found in toolboxes.
- To serve as factories for making objects (which contain variables and methods from the non-static parts of class definitions).
**Question 4:**
The subclass inherits the superclass's structure and behavior. That is, just by extending a class (with "extends") a subclass automatically has the same variables and subroutines (methods) that the superclass has. This is called inheritance. The subclass may then add to these variables and methods, and can even redefine some of them.

Example could be:

Mammal = superclass  
Dog = subclass and Horse = subclass

**Question 5:**
This can be done with two statements:

```java
Account myAccount;
myAccount = new Account(500.00);
```

Or it can be done with a single statement:

```java
Account myAccount = new Account(500.00);
```

**Question 6:**
```java
public class Point3D extends Point2D {
    public double z;
}
```

**Question 7:**
The program will emit the following:

```
x[0] = 0
x[1] = 0
x[2] = 3
x[3] = 100
x[4] = 0
```

**Question 8:**
```java
public class PairOfDice {
    private int die1;   // Number showing on the first die.
    private int die2;   // Number showing on the second die.

    public PairOfDice() {
        // Constructor. Rolls the dice, so that they initially
        // show some random values.
        roll();   // Call the roll() method to roll the dice.
    }

    public void roll() {
        // Roll the dice by setting each of the dice to be
        // a random number between 1 and 6.
    }
}
```
```java
die1 = (int)(Math.random()*6) + 1;
die2 = (int)(Math.random()*6) + 1;
}
public int getDie1() {
    // Return the number showing on the first die.
    return die1;
}
public int getDie2() {
    // Return the number showing on the second die.
    return die2;
}
public int getTotal() {
    // Return the total showing on the two dice.
    return die1 + die2;
}
}  // end class PairOfDice
```