CSCI2010 Principle of Computer Science

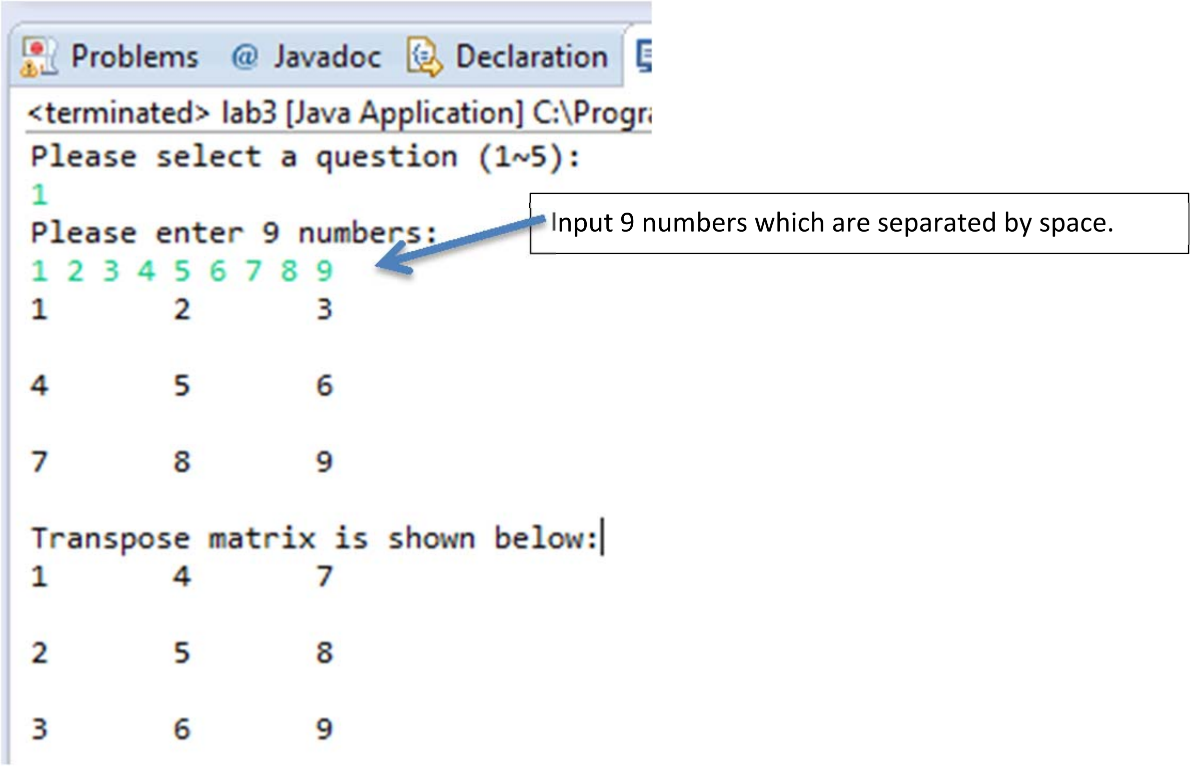
Laboratory Three

Activity 0: Prepare project under eclipse

- Create a project called “lab3”;
- Copy “lab3.java” into your workspace;
- Copy “questions.java” into your workspace;
- Right click project “lab3” and select “refresh”;

Activity 1: Transposing a Matrix (1 mark)

In q1, write a program that asks the user to enter 9 integers and place them into a 3x3 2d array. Then transpose the array so the first row becomes the first column, the second row becomes the second column, and the third row becomes the third column. Here is an example:

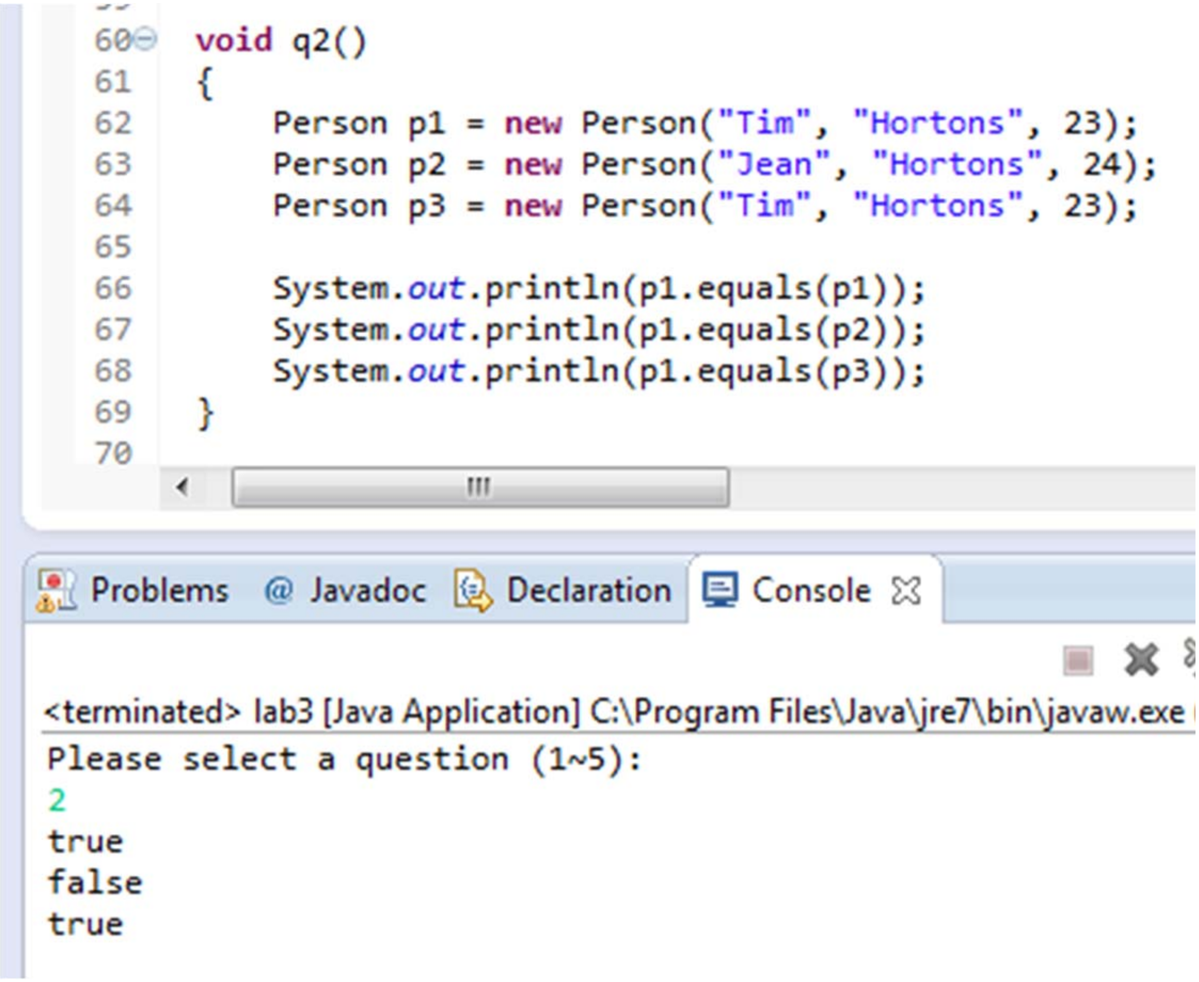


```
<terminated> lab3 [Java Application] C:\Progr
Please select a question (1~5):
1
Please enter 9 numbers:
1 2 3 4 5 6 7 8 9
1       2       3

4       5       6

7       8       9

Transpose matrix is shown below:
1       4       7

2       5       8

3       6       9
```

Input 9 numbers which are separated by space.

Activity two: Overriding Methods (2 marks)

You can find “Person.java” in lab3 package, it is a simple class that represents a person. Add it into your lab3 project. Since it does not explicitly inherit from any other classes, it inherits from the Object class in java.lang by default. This means it inherits all of Object’s methods, including the *equals* method. However, the *equals* method in Object only returns true if both objects are actually the same object in memory, aliases of one another. Override the *equals* method so it returns true if two different *Person* objects share the same instance data. The following snapshot is an expected output.



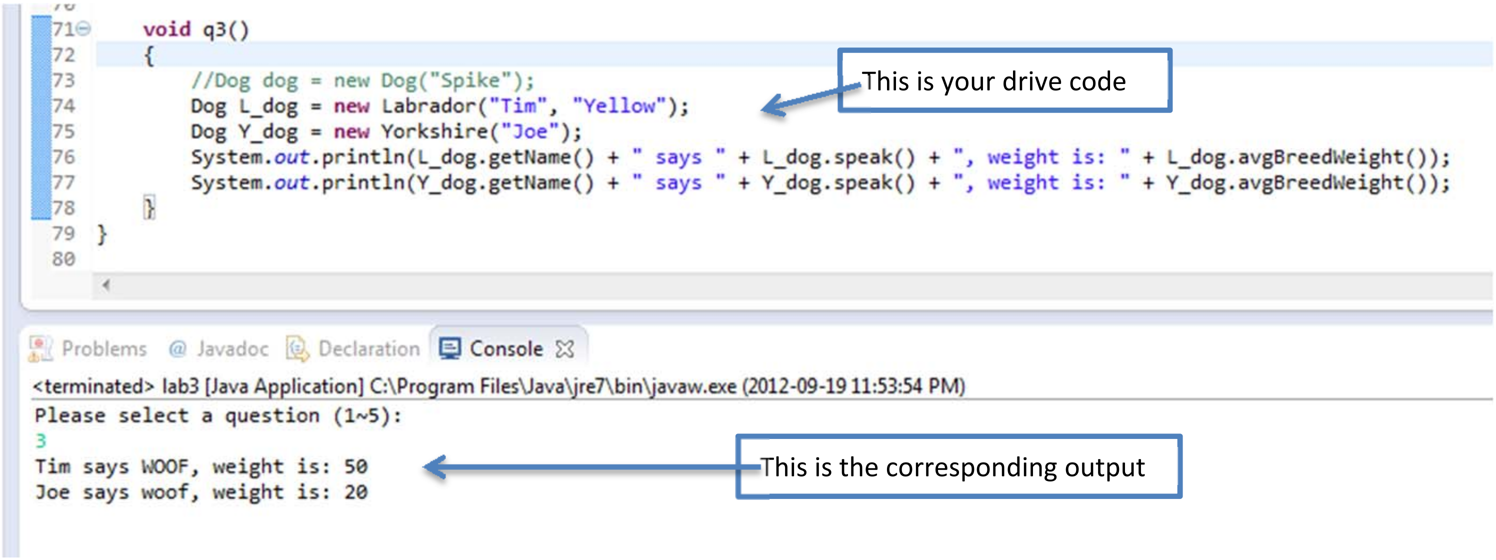
```
void q2()
{
    Person p1 = new Person("Tim", "Hortons", 23);
    Person p2 = new Person("Jean", "Hortons", 24);
    Person p3 = new Person("Tim", "Hortons", 23);

    System.out.println(p1.equals(p1));
    System.out.println(p1.equals(p2));
    System.out.println(p1.equals(p3));
}
```

```
<terminated> lab3 [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe
Please select a question (1~5):
2
true
false
true
```

Activity 3: Inheritance (2 marks)

There are three class files called “Dog.java”, “Yorkshire.java” and “Laborador.java” in lab3 package, now you can add them into your project and refresh your project.

1) Now, you should see two errors in “Yorkshire.java” and “Laborador.java” respectively. Try to fix them.
2) Add an abstract function called “avgBreedWeight” which return an integer number to the Dog class, you should see three errors in Dog.java, Yorkshire.java and Laborador.java. Fix these errors.
3) Define two Dog instance in q3, one is “Yorkshire”, another is “Laborador”. Call corresponding speak function for two instance and print out the result. Code example is shown below:



```
    void q3()
    {
        //Dog dog = new Dog("Spike");
        Dog L_dog = new Labrador("Tim", "Yellow");
        Dog Y_dog = new Yorkshire("Joe");
        System.out.println(L_dog.getName() + " says " + L_dog.speak() + ", weight is: " + L_dog.avgBreedWeight());
        System.out.println(Y_dog.getName() + " says " + Y_dog.speak() + ", weight is: " + Y_dog.avgBreedWeight());
    }
}
```

This is your drive code

```
<terminated> lab3 [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (2012-09-19 11:53:54 PM)
Please select a question (1~5):
3
Tim says WOOF, weight is: 50
Joe says woof, weight is: 20
```

This is the corresponding output

Activity 4: Morphing Dogs (1 marks)

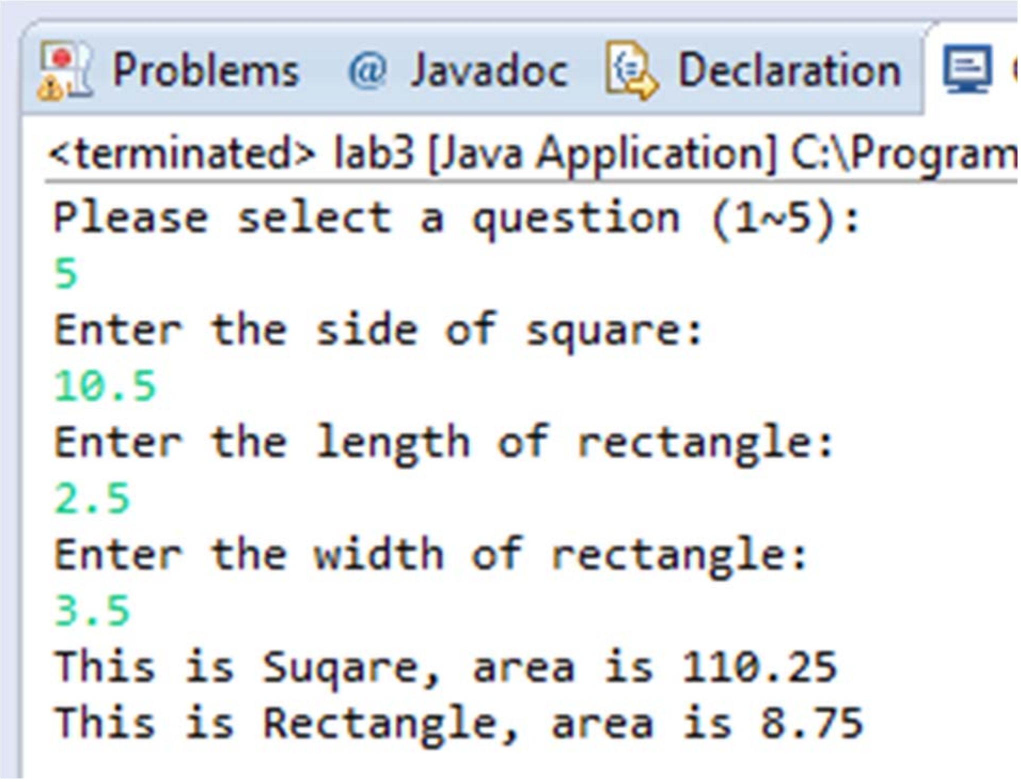
q4 is an uncompleted function. Fill in the missing code so that every time the loop is executed, the user decides what type of dog myDog will be.

Activity 5: (4 marks)

These coding exercises reinforce the lessons learned in the lectures and provide programming experience in Java. For each of the following problems, write a program or a program segment that performs the specified action.

1. Write an empty class declaration for an abstract class called *Shape*.
2. In the class from Step 1, create a protected instance variable “shapeName” of type String, and write an accessory method “*getName*” for obtaining its value.
3. In the class of Step 2, define an abstract method *getArea* that returns a double representation of a specific shape’s area. Subclasses of this class must implement *getArea* to calculate a specific shape’s area.
4. Define a class *Square* that inherits from class *Shape* from step 3; it should contain an instance variable *side,* which represents the length of a side of the square. Provide a constructor that takes one argument representing the side of the square and sets the side variable. Ensure that the side is greater than or equal to 0. The constructor should set the inherited *shapeName* variable to the string "*Square*".
5. The *Square* class from step 4 should implement the *getArea* method of its abstract superclass; this implementation should compute the area of the square and return the result.
6. Define a class *Rectangle* that inherits from class *Shape* of step 3. The new class should contain instance variables *length* and *width*. Provide a constructor that takes two arguments representing the length and width of the rectangle, sets the two variables and sets the inherited *shapeName* variable to the string "*Rectangle*". Ensure that the length and width are both greater than and equal to 0.
7. The Rectangle class from step 6 should also implement the *getArea* method of its abstract superclass; this implementation should compute the area of the rectangle and return the result.
8. In q5, write an application that tests the *Square* and *Rectangle* classes from step 5 and 7, respectively. Create an array of type *Shape* that holds an instance of *Square* and an instance of *Rectangle*. The program should polymorphically compute and display the areas of both objects. Allow a user to enter the values for the side of the square and the length and width of the rectangle.

An example output is shown below:



```
<terminated> lab3 [Java Application] C:\Program
Please select a question (1~5):
5
Enter the side of square:
10.5
Enter the length of rectangle:
2.5
Enter the width of rectangle:
3.5
This is Suqare, area is 110.25
This is Rectangle, area is 8.75
```

What need to be submitted?

Please submit all java source code files on Black board.

- lab3.java
- questions.java
- Person.java
- Dog.java
- Labrador.java
- Yorkshire.java
- Shape.java
- Square.java
- Rectangle.java