Lab 1

Classes and Objects

1. Basic classes and objects. Based on Gaddis Algorithm Workbench 6.3. (10 points)

Create a Java project with two class files. Please remember to type in this code rather than copy-and-pasting; Word loves to mangle source code.

MySquare.java:

// Student Name Today's Date

**public** **class** MySquare {

**private** **double** sideLength;

**public** **double** getArea() {

**return** sideLength \* sideLength;

}

**public** **double** getSideLength() {

**return** sideLength;

}

**public** **void** setSideLength(**double** l) {

sideLength = l;

}

}

ClassTest.java:

// Student Name Today's Date  
**public** **class** ClassTest {  
 **public** **static** **void** main(String[] args) {

MySquare X = **new** MySquare();

X.setSideLength(5.0);

System.***out***.println(

" A square with sides of length " + X.getSideLength() +

" has area " + X.getArea() + ".");

}

}

Run the code. When ClassTest.Java runs as expected, make the following changes:

* 1. Add a new variable Y of type MySquare.
  2. Set the side length of variable Y to 3.5.
  3. Use the getArea() method of Y to calculate the area of a square with side length 3.5
  4. Display the area calculated in 1.3 on the screen using println.
  5. Create code to calculate and display the area of squares with side length 45.0, 52.0, and 13.5.

Rubric:

Student name and today’s date is a comment in the first line of the programs: -5 points if fails  
Predicted output listed on lab: -5 points if fails

Screenshot and output of a successful program run: 2 points

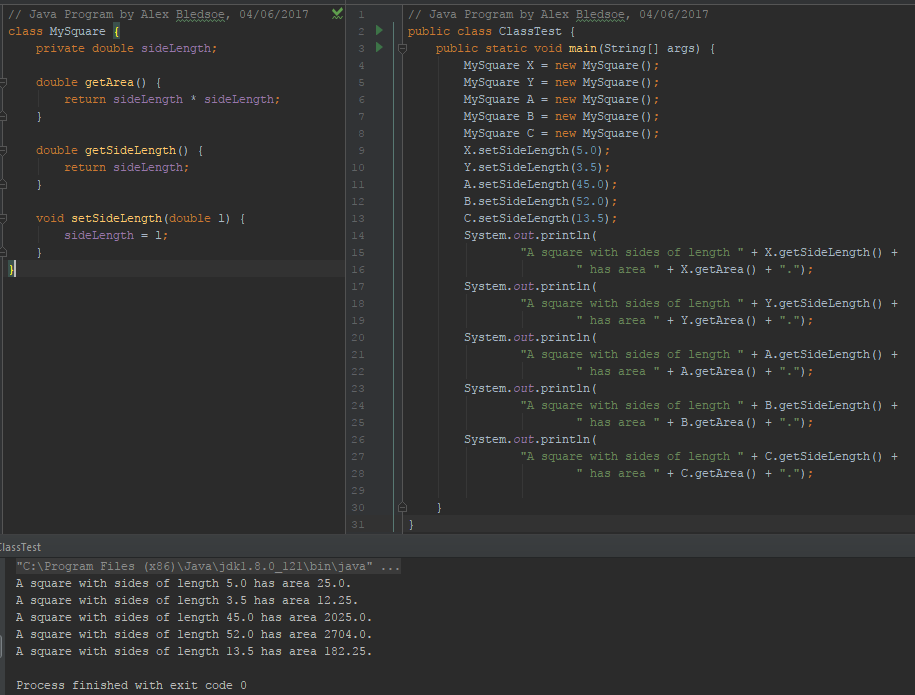
MySquare.java code included: 1 points

ClassTest.java:

Square Y with side length 3.5, calculated and output area: 2 points

Additional calculations for side length 45.0, 52.0, and 13.5: 2 points each, total of 6 points

Please paste a screenshot of a successful program run, and copy-and-paste the source code from **both** .java files here:



**MySquare.java:**

// Java Program by Alex Bledsoe, 04/06/2017  
class MySquare {  
 private double sideLength;  
  
 double getArea() {  
 return sideLength \* sideLength;  
 }  
  
 double getSideLength() {  
 return sideLength;  
 }  
  
 void setSideLength(double l) {  
 sideLength = l;  
 }  
}

**ClassTest.java**

// Java Program by Alex Bledsoe, 04/06/2017  
public class ClassTest {  
 public static void main(String[] args) {  
 MySquare X = new MySquare();  
 MySquare Y = new MySquare();  
 MySquare A = new MySquare();  
 MySquare B = new MySquare();  
 MySquare C = new MySquare();  
 X.setSideLength(5.0);  
 Y.setSideLength(3.5);  
 A.setSideLength(45.0);  
 B.setSideLength(52.0);  
 C.setSideLength(13.5);  
 System.*out*.println(  
 "A square with sides of length " + X.getSideLength() +  
 " has area " + X.getArea() + ".");  
 System.*out*.println(  
 "A square with sides of length " + Y.getSideLength() +  
 " has area " + Y.getArea() + ".");  
 System.*out*.println(  
 "A square with sides of length " + A.getSideLength() +  
 " has area " + A.getArea() + ".");  
 System.*out*.println(  
 "A square with sides of length " + B.getSideLength() +  
 " has area " + B.getArea() + ".");  
 System.*out*.println(  
 "A square with sides of length " + C.getSideLength() +  
 " has area " + C.getArea() + ".");  
  
 }  
}

2. Employee Class. Based on Gaddis Programming Challenges 6.1. (10 points)

Write a class named Employee that has the following fields:

• name. The name field references a String object that holds the employee’s name.

• idNumber. The idNumber is an int that holds the employee’s ID number.

• department. The department field is a String that holds the name of

the department where the employee works.

• position. The position field is a String that holds the employee’s

job title.

Access the fields using variableName.fieldName. For example:

{  
 Employee e1 = new Employee;  
 e1.name = “Susan Meyers”;  
 e1.idNumber = 47899;  
 // and similar for department and position  
}

Once you have written the class, write a separate (main) program that creates three Employee objects to hold the following data:

Name ID Number Department Position

Patel Koresh 432 Accounting CEO

Smidge Crossbow 391 IT CIO

Joy Temple 814 Manufacturing Shift lead

The program should store this data in the three objects and then display the data for each employee on the screen.

Please remember to put your name and date of program completion at the top of **both** class files.

Rubric:  
Student name and today’s date is a comment in the first line of the programs: -5 points if fails  
Employee class defined correctly: 3 points  
Main program creates Employee variables correctly: 1 points  
Three Employee variables with correct data: 3 points  
Output from three employee variables: 3 points

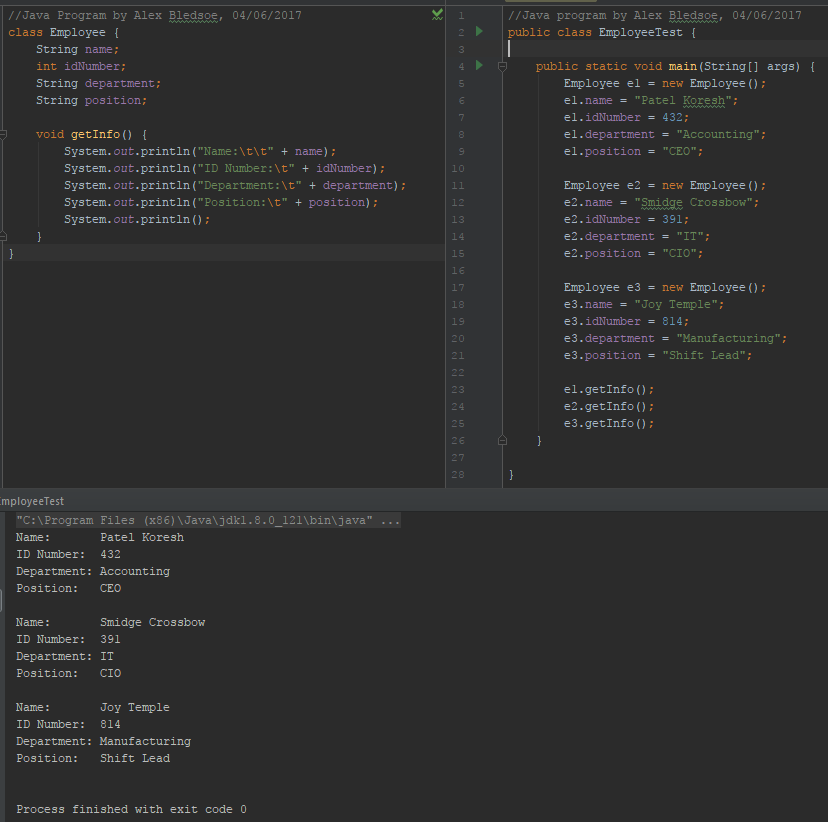
Please paste a screenshot of a successful program run, and copy-and-paste the source code from **both** .java files, here:

**Employee.java:**

//Java Program by Alex Bledsoe, 04/06/2017  
class Employee {  
 String name;  
 int idNumber;  
 String department;  
 String position;  
  
 void getInfo() {  
 System.*out*.println("Name:\t\t" + name);  
 System.*out*.println("ID Number:\t" + idNumber);  
 System.*out*.println("Department:\t" + department);  
 System.*out*.println("Position:\t" + position);  
 System.*out*.println();  
 }  
}

**EmployeeTest.java:**

//Java program by Alex Bledsoe, 04/06/2017  
public class EmployeeTest {  
  
 public static void main(String[] args) {  
 Employee e1 = new Employee();  
 e1.name = "Patel Koresh";  
 e1.idNumber = 432;  
 e1.department = "Accounting";  
 e1.position = "CEO";  
  
 Employee e2 = new Employee();  
 e2.name = "Smidge Crossbow";  
 e2.idNumber = 391;  
 e2.department = "IT";  
 e2.position = "CIO";  
  
 Employee e3 = new Employee();  
 e3.name = "Joy Temple";  
 e3.idNumber = 814;  
 e3.department = "Manufacturing";  
 e3.position = "Shift Lead";  
  
 e1.getInfo();  
 e2.getInfo();  
 e3.getInfo();  
 }  
  
}



3. Comprehensive – Dice Roll Simulator, based on Gaddis programming challenge 6.16. (10 points)

Write a class named Die. The Die class should have the following field:

• A private integer named currentNumber. The currentNumber field will hold an integer between 1-6. indicating the side of the die that is facing up.

The Die class should have the following methods:

• A no-arg constructor that randomly determines which side is up and sets currentNumber accordingly.

• A void method named roll that simulates rolling the die. When the roll

method is called, it randomly determines the number that is shown on the die, and sets currentNumber accordingly.

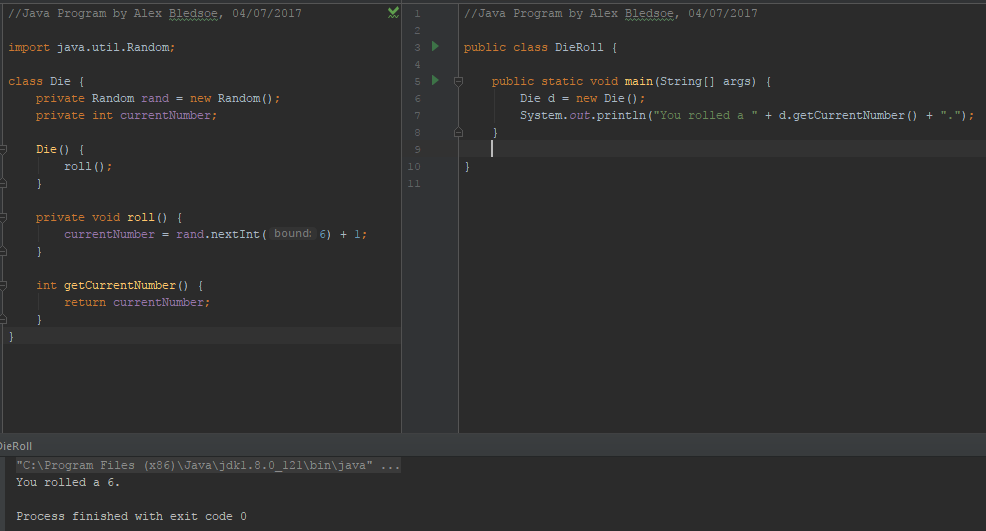
• A method named getCurrentNumber that returns the value of the currentNumber field.

Write a program that demonstrates the Die class. The program should create an instance of

the class and display the side that is initially facing up

Rubric:  
Student name and date is in a comment on the first line of the programs: -5 points if fails  
Die class generates numbers 1-6 on repeated runs: -5 points if fails  
currentNumber field is private: -5 points if fails  
Die class definition, methods, constructors: 5 points  
Main program instantiates one die and displays value: 5 points

Please paste a screenshot of a successful program run, and copy-and-paste the source code from both .java files, here:



**Die.java:**

//Java Program by Alex Bledsoe, 04/07/2017  
  
import java.util.Random;  
  
class Die {  
 private Random rand = new Random();  
 private int currentNumber;  
  
 Die() {  
 roll();  
 }  
  
 private void roll() {  
 currentNumber = rand.nextInt(6) + 1;  
 }  
  
 int getCurrentNumber() {  
 return currentNumber;  
 }  
}

**DieRoll.java:**

//Java Program by Alex Bledsoe, 04/07/2017  
  
import java.util.Random;  
  
class Die {  
 private Random rand = new Random();  
 private int currentNumber;  
  
 Die() {  
 roll();  
 }  
  
 private void roll() {  
 currentNumber = rand.nextInt(6) + 1;  
 }  
  
 int getCurrentNumber() {  
 return currentNumber;  
 }  
}

4. Comprehensive – Dice Roll Simulator, based on Gaddis programming challenge 6.16. (15 points)

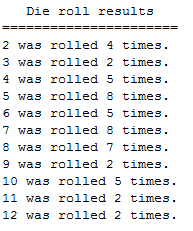
Write a program that demonstrates the Die class from the previous problem.

This program will roll **two** Die objects repeatedly, add the value from the two dies together, to display a total for that roll.

Use a loop to do this 50 times. This simulates 50 independent rolls of two dice.

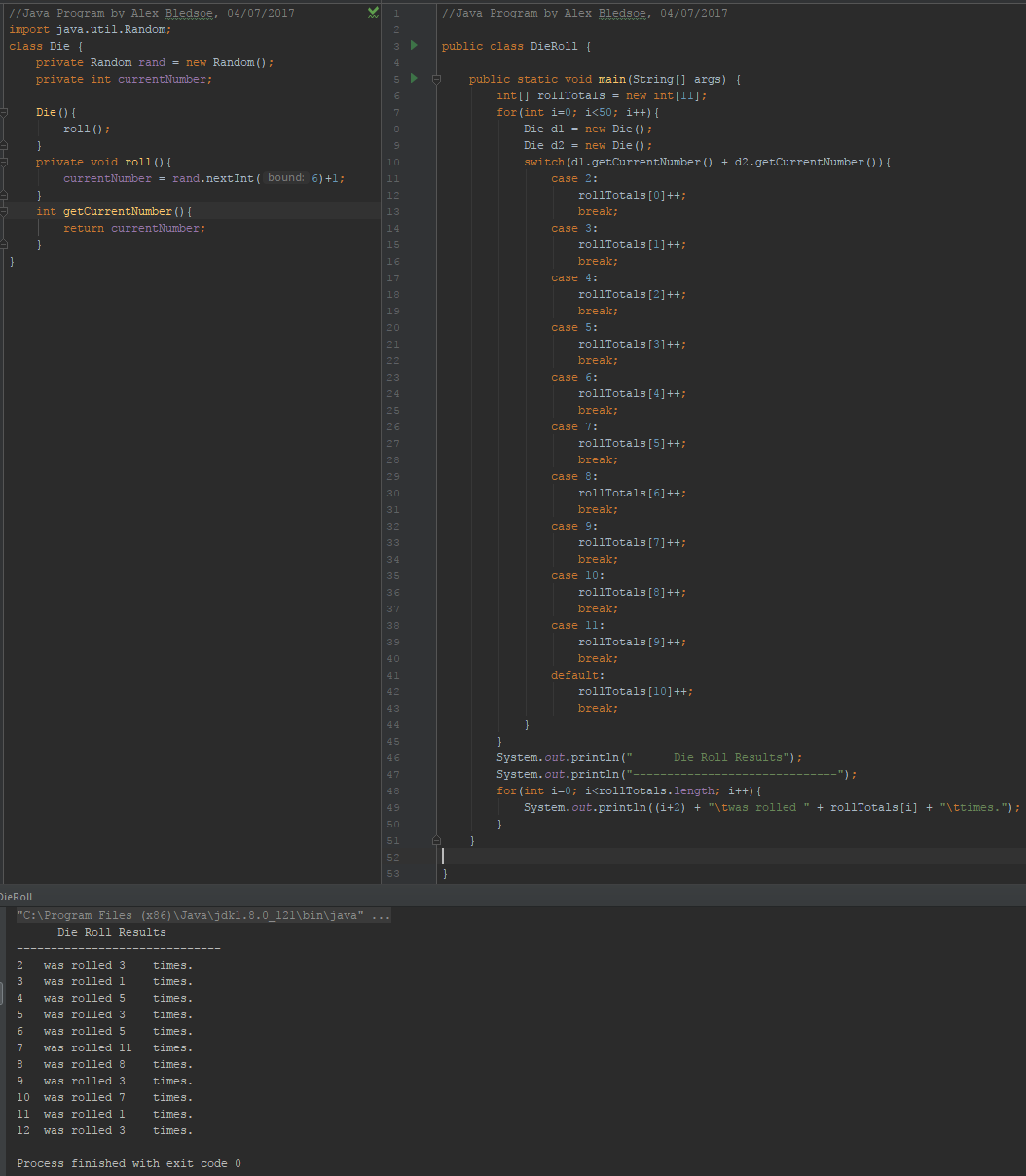
The program should keep track of how many times each total value (from 2 - 12) appears, and output the totals at the end.

Program output *could* look like this. The formatting is not required, but the content is.



Rubric:  
Student name and date is in a comment on the first line of the programs: -5 points if fails  
All values 2-12 occur in the totals: -10 if fails  
Main program loop die toll 50 times: 5 points  
Main program count outcome for totals: 10 points

Please paste a screenshot of a successful program run, and copy-and-paste the source code from both .java files, here:



**Die.java:**

//Java Program by Alex Bledsoe, 04/07/2017  
  
import java.util.Random;  
  
class Die {  
 private Random rand = new Random();  
 private int currentNumber;  
  
 Die() {  
 roll();  
 }  
  
 private void roll() {  
 currentNumber = rand.nextInt(6) + 1;  
 }  
  
 int getCurrentNumber() {  
 return currentNumber;  
 }  
}

**DieRoll.java:**

//Java Program by Alex Bledsoe, 04/07/2017  
  
public class DieRoll {  
  
 public static void main(String[] args) {  
 int[] rollTotals = new int[11];  
 for (int i = 0; i < 50; i++) {  
 Die d1 = new Die();  
 Die d2 = new Die();  
 switch (d1.getCurrentNumber() + d2.getCurrentNumber()) {  
 case 2:  
 rollTotals[0]++;  
 break;  
 case 3:  
 rollTotals[1]++;  
 break;  
 case 4:  
 rollTotals[2]++;  
 break;  
 case 5:  
 rollTotals[3]++;  
 break;  
 case 6:  
 rollTotals[4]++;  
 break;  
 case 7:  
 rollTotals[5]++;  
 break;  
 case 8:  
 rollTotals[6]++;  
 break;  
 case 9:  
 rollTotals[7]++;  
 break;  
 case 10:  
 rollTotals[8]++;  
 break;  
 case 11:  
 rollTotals[9]++;  
 break;  
 default:  
 rollTotals[10]++;  
 break;  
 }  
 }  
 System.*out*.println(" Die Roll Results");  
 System.*out*.println("------------------------------");  
 for (int i = 0; i < rollTotals.length; i++) {  
 System.*out*.println((i + 2) + "\twas rolled " + rollTotals[i] + "\ttimes.");  
 }  
 }  
  
}