**Week 5 Mon-Tue-Wed Lab Assignments**

1. Write a Java program that will read in course score (a value between 0 and 100), it determines a student’s course grade. This program will determine the course grade based on the course score – using three methods: main, getScore and printGrade. A Wrapper class called DoubleClass will be given to use along with your program. setNum() and getNum() methods are used from the DoubleClass.

[ComputerTheGrade\DoubleClass.java](ComputerTheGrade/DoubleClass.java)

**mian() method**: will get the course score and display the course grade.

**getScore() method:** will get the user input and print the course score.

**printGrade() method:** will calculate the course grade and print the course grade.

**In main(),** you will be calling getScore() method, passing in the DoubleClass object. Then call the printGrade() method, passing in DoubleClass getNum() method.

**In getScore()** method your parameter will be the DoubleClass object (reference variable). User will enter in the course score, save the course score then call the DoubleClass setNum() method and display the course score.

**In printGrade()**, using nested if….else, compare the score entered by the user and display appropriate letter grade.

**IPO chart, algorithm, code and comments required for full credit.**

**Expected Results:**

Based on the course score, this program computes the course letter grade.

Enter the course score: 90.50

The course score is 90.50

Your letter grade for the course is A

1. This java program will pass in a String object as parameter to a method and change the passed string.

Within main() method create a string variable containing a string, for example, “Hello”. Then display the string to the console, call the stringParameter() method and pass in the string you have created. Then display the str once again after the stringParameter() method call.

Within the stringParameter() method, display the original string passed from main(), then change the string from “Hello” to “Sunny Day”, then display the changed string.

**Expected Results:**

String value before calling the method stringParameter: Hello

In the method stringParameter

Passed String value before changing its value: Hello

Passed String value after changing its value: Sunny Day

String value after calling the method stringParameter: Hello

1. **Self-Test: Consider the following method:**

public static int mystery (int ex, double y, char ch)

{

int u;

if(‘A’ <= ch && ch <= ‘R’)

return (2 \* x + (int) (y));

else

return((int) (2 \* y) – x);

}

**What is the output of the following Java statements?**

* 1. System.out.println(mystery(5, 4.3, ‘B’));
  2. System.out.println(mystery(4, 9.7, ‘v’));
  3. System.out.println(2 \* mystery(6, 3.9, ‘D’));

1. **In the following program, number the marked statements to show the order in which they will execute (the logical order of execution).**

import java, util.\*;

public class Exercise6

{

static Scanner console = new Scanner(System.in);

public static void main(String[] args)

{

int num1;

int num2;

\_\_\_\_\_\_\_\_System.out.println(“Please enter two integers on” +

separate lines”);

\_\_\_\_\_\_\_\_num1 = console.nextInt();

\_\_\_\_\_\_\_\_num2 = console.nextInt();

\_\_\_\_\_\_\_\_func(num1, num2);

\_\_\_\_\_\_\_\_System.out.println(“The two integers are “ + num1 +

“, “ + num2);

}//end of main()

public static void func(int val1, int val2)

{

int val3;

int val4;

\_\_\_\_\_\_\_\_val3 = val1 + val2;

\_\_\_\_\_\_\_\_val4 = val1 \* val2;

\_\_\_\_\_\_\_\_System.out.println(“The sum and product are “ + val3 +

“ and “ + val4);

}//end of func()

}//end of class

1. Given the Illustrate class, write a Demo or Test program that will instantiate objects of Illustrate class and call its static class members.

**In the Demo class, called StaticMembers.**

Instantiate 2 objects of the Illustrate class. For object 1 pass in 3 and for object 2 pass in 5 -- in the constructor. Call the incrementY() method at class level then increment the count variable from Illustrate class by 1. Display the object1 and object 2. Now using the object1 name call the increment() method, and setX to 8 for object 1. Print the Object1 and Object2. Now using the object2 name call increment() method and setX to 23 for object 2. Print the Object1 and Object2.

**Code and comments required for full credit. And don’t forget to include the given class (Illustrate) and the Demo class (StaticMembers) --- also place both files into one folder then compile and test the classes.**

**Sample Run:**

illustrateObj1: x = 3, y =1, count =1

illustrateObj2: x =5, y =1, count =1

\*\*\*\*Increment y using illustrateObj1\*\*\*\*

illustrateObj1: x = 8, y =2, count =1

illustrateObj2: x =5, y =2, count =1

\*\*\*\*Increment y using illustrateObj2\*\*\*\*

illustrateObj1: x = 8, y =3, count =1

illustrateObj2: x =23, y =3, count =1

1. **Mark the following statements as True or False:**
   1. The instance variables of a class must be of the same type.
   2. The methods of a class must be public.
   3. A class can have more than one constructor.
   4. A constructor can return a value of the int type.
   5. An accessor method of a class access and modifies the data members of the class.
2. **Self-Test: Consider the definition of the following class:**

**public class CC**

**{**

**private int u;**

**private int v;**

**private double w;**

**public CC() Line 1**

**{}**

**public CC(int a) Line 2**

**{}**

**public CC(int x, int b) Line 3**

**{}**

**public CC(int a, int b, double d) Line 4**

**{}**

**}**

1. Give the line number containing the constructor that is executed in each of the following declaration:
2. CC one = new CC();
3. CC two = new CC(5, 6);
4. CC three = new CC(2, 8, 3.5);
5. Write the definition of the constructor in Line 1 so that the instance variables are initialized to 0.
6. Write the definition of the constructor in Line 2 so that the instance variable u is initialized according to the value of the parameter, and the instance variable v and w are initialized to 0.
7. Write the definition of the constructor in Line 3 so that the instance variables u and v are initialized according to the values of the parameters a and b, respectively, and the instance variable w is initialized to 0.0.
8. Write the definitions of the constructors in Line 4 so that the instance variables u, v, and w are initialized according to the values of the parameters a, b, and d, respectively.
9. **Self-Test: Write the definition of a class that has the following properties:**
   1. The name of the class is **Secret**.
   2. The **class Secret** has four instance variables: **name** of type String, **age** and **weight** of type int, and **height** of type double.
   3. **The class Secret has the following methods:**
      1. **print** ---- outputs the data stored in the data members with the appropriate titles.
      2. **setName** ---- method to set the name
      3. **setAge** ---- method to set the age
      4. **setWeight** ---- method to set the weight
      5. **setHeight** ---- method to set the height
      6. **getName** ---- value returning method to return the name
      7. **getAge** ---- value returning method to return the age
      8. **getWeight** ---- value returning method to return the weight
      9. **getHeight** ---- value returning method to return the height
      10. **default constructor** ---- the default value of name is the empty string “”; the default values of age, weight, and height are 0.
      11. **constructor with parameters** --- sets the value of the instance variables name, age, weight, and height to the values specified by the user.
   4. Write the definitions of the method members of the class Secret, as described in Part C.