**LAB Grade:**

|  |  |
| --- | --- |
| http://www.aus.edu/common/images/logo_small.gif | **American University of Sharjah**  **College of Engineering**  **Department of Computer Science and Engineering** |

**Faculty Details**

|  |  |
| --- | --- |
| Instructor: Lab Instructor :  Office:  Phone:  e-mail:  Semester: | Dr. Michel Pasquier Mr. Suresh Radder  EB2-126A  971-6-515-2924 [sradder@aus.edu](mailto:sradder@aus.edu) Spring 2017 |

**Course Details**

|  |  |
| --- | --- |
| Course:  Semester: | CMP 256 L – GUI Design and Programming  Spring 2017 |

**Lab and Assignment Details**

|  |  |
| --- | --- |
| Assignment No:  Assignment Topic:  Date:  Lab Location: | 1  Introduction to Netbeans and Java Basics.  31st Jan 2017  EB2 -125 |

**Academic Integrity Pledge**

|  |
| --- |
| As a student of American University of Sharjah, I here by state that I will abide by the AUS Integrity Pledge that:   * I will hold myself accountable for all that I say and write. * I will hold myself responsible for the academic integrity of my work * I will not carry out unauthorized copying or printing of the work of others * I will not misrepresent my work nor give or receive unauthorized aid * I will behave in a manner that demonstrates concern for the personal dignity, rights and freedoms of all members of the community * I will respect university property and the property of others; and * I will not tolerate a lack of respect for these values.   **Student Name:**  **Student ID:** |

**CMP 256L – GUI Design and Programming**

**Lab 1 - Java Basics and Netbeans**

***Objectives***

* Introduction to Java and the NetBeans IDE
* Write and test simple Java programs
* Understand the main components of Java programs

***Exercise 1***

Create a Java application that prompts the user to enter ***n*** numbers (doubles) and finds the average of ***n***numbers and prints the average value. The value of ***n*** must also be input by the user.

*References:*

*NetBeans IDE Java Quick Start Tutorial:* <https://netbeans.org/kb/docs/java/quickstart.html#run>

*How to use Scanner class:* <http://forums.hotjoe.com/posts/list/128.page>

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication4;

import java.util.Scanner;

/\*\*

\*

\* @author g00061542

\*/

public class JavaApplication4 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

System.out.println("Please enter n numbers.");

Scanner in=new Scanner(System.in);

double n=in.nextInt();

double sum;

sum=0;

System.out.println("please enter the values");

for(int x=0;x<n;x++)

{

double v=in.nextInt();

sum=sum+v;

}

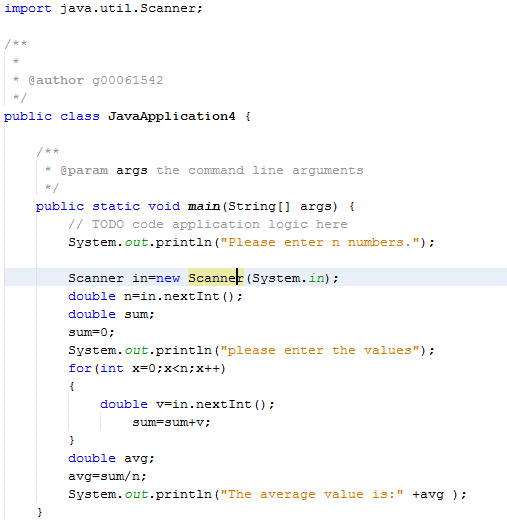
double avg;

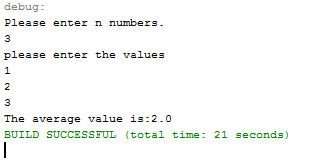
avg=sum/n;

System.out.println("The average value is:" +avg );

}

}





***Exercise 2***

Write a Java program that prompts the user for the radius of a sphere and prints the *area* and the *volume* of the sphere.

*Hints:* Area of sphere *= ,* Volume of Sphere*=* and use *PI a*nd *pow* functions provided by the *java.lang.Math*  class.

package javaapplication4;

import java.util.Scanner;

import java.lang.Math;

public class JavaApplication4 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

System.out.println("please enter desired radius: ");

Scanner in=new Scanner(System.in);

double radius=in.nextInt();

double area;

area=4\*Math.PI\*Math.pow(radius,2);

System.out.println("Area of sphere = "+area);

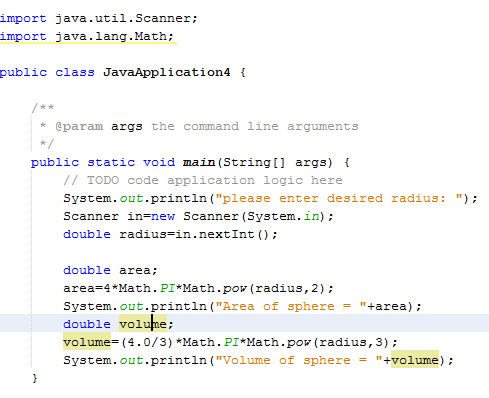
double volume;

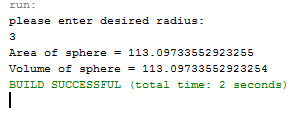
volume=(4.0/3)\*Math.PI\*Math.pow(radius,3);

System.out.println("Volume of sphere = "+volume);

}

}





***Exercise 3***

Playing cards consist of 52 cards categorized in four categories (Diamonds, Spades, Hearts and Clubs). Each category consists of 13 cards, ranging from 2 to 10, then Jack (11), Queen (12), King (13) and Ace(1).

Write a java program that deals 13 cards for a player. A card should not be served more than once; use a two-dimensional array of ***integers*** of four rows and 13 columns (initialized to zero) to know if a card was served previously or not.

Serving the cards should be done randomly, you can generate two random numbers one to represent the category while the other represents the card number. Once a card is served you must mark its corresponding location in the two-dimensional array to *1* (given that it was not served previously of course).

Finally, the program prints the cards dealt to the player,

*Sample Output:*

C2 C3 C6 C8 CK

D4 D8

H3

SA S3 S4 S5 SQ

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication4;

import java.util.Arrays;

import java.util.Random;

public class JavaApplication4 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

final int RMax=4;

final int CMax=13;

int[][] card=new int [RMax][CMax];

for(int x=0;x<card.length;x++)

{

for(int j=0;j<card[x].length;j++)

{

card[x][j]=0;

}

}

Random v=new Random();

int category =v.nextInt(4);

int cardNumb=v.nextInt(13);

for(int x=0;x<card.length;x++)

{

for(int j=0;j<card[x].length;j++)

{

if (card[x][j]==0)

{

System.out.println(category);

System.out.println(cardNumb);

}

else

{

card[x][j]=1;

}

}

}

System.out.println(Arrays.deepToString(card));

}

}

***Hand in:*** Solution code with sample input/output

***Due Date:*** Before the beginning of next lab session ends.

***Grading policy:*** Mentioned in the course outline posted on ilearn.