Alan Palayil

Anwar Adam

Rezwan Rifat

**CS 115 Fall 2019 Lab #4**

Due: **Monday, September 23rd, 5:00 PM**

Points: **20**

**Instructions:**

1. Use this document template to report your answers. Enter all lab partner names at the top of the first page.

2. You don’t need to finish your lab work during the corresponding lab session.

3. ZIP your lab report and java files (if any) into a single ZIP file. Name the ZIP file as follows:

LastName\_FirstName\_CS115\_Lab4\_Report.zip

4. Submit the final document to Blackboard Assignments section before the due date. No late submissions will be accepted.

5. ALL lab partners need to submit a report, even if it is the same document.

**Objectives:**

1. (20 points) Use classes and objects.

**Problem 1 [20 points]:**

For each of the following four problems complete an 1.Input-Process-Output Design, 2.Test Table, 3.Java Program.

1. Input-Process-Output Design (answer the questions below, fill the inputs and outputs table, and come up with pseudocode) **[1 out of 5 points each]**,

2. Test Plan Table (fill the Test plan table as for the previous lab) **[1 out of 5 points each],**

3. Java Program - Implement your Input-Process-Output Design in java. Use Scanner for user input. make sure to format your output to correct number of decimal places, if applicable. **[3 out of 5 points each]**.

INPUTS: What are the inputs?

n What format / data type are they? (integer, real number, single character, string - a sequence of characters)

n Any valid / invalid / illegal / special values? (positive, negative, valid range, etc.)

n How do you get them? (enter manually, ask user, read from file, etc.)

PROCESS: How do you get from inputs to the outputs you want?

n What are the calculation steps?

n To follow these steps, what else do you need? (formulas, etc.)

n Other variables, constants, conversions (besides input and output variables)

OUTPUTS: What are the outputs?

n What format / data type are they in? (integer, floating-point, character, or string)

n Any valid / invalid / illegal / special values? (positive, negative, valid range, etc.)

n How do you output them? (display on screen, save to a file, plot, tabularize, etc.)

==================================================================

A. Prompt the user for the length of the side of a cube, determine the radius of a sphere that has the same surface area as that cube and then the radius of a sphere that has the same volume as that cube. Output both radii (to 2 decimal places) with messages. In your test table be sure to test a range of values and types of values. **[5 points].**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Inputs and outputs (use “N/A”, “undefined”, “none”, etc. If necessary)** | | | | | |
| Variable .  name | Input or Output? | Data type / format | Constraints | Special cases | Comments |
| Sides (s) | Input | float | s>0 | s>0 | Side length of a cube is 5 |
| Radius of surface area (rs) | Input & output | float | Rs> 0 | Rs> 0 | As it is input again after output to find for the sphere |
| Radius of volume (rv) | Input & output | float | rv>0 | rv>0 | As it is input again after output to find for the sphere |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |
| --- |
| **Pseudocode:** |
| GET length of Side of cube  COMPUTE Surface Area of Cube, Volume of Cube  Surface Area=(Length\*Length)\*6  Volume=Length\*Length\*Length  STORE Surface Area and Volume  COMPUTE Radius for same surface Area=(Surface Area of Cube)/4π  Radius for same Volume=∛{(Volume of cube\*3)/4π}  STORE Radius for Surface Area, Radius for volume  DISPLAY Radius for Surface Area, Radius for volume |

|  |  |  |
| --- | --- | --- |
| **Test case name (ex. “negative height”, “typical conditions”, etc.)** | **Input data set for this test case** | **Explain why you chose this test case** |
| Control Test | Length=5.00, Width=5.00,Height=5.00 | To compare answer to other test cases |
| One Negative Input | Length=-5.00, Width=5.00,Height=5.00 | To see it shows error as answer cannot be negative |
| Two Negative Input | Length=-5.00, Width=5.00,Height=-5.00 | To see if it shows error as answer is correct, but input is still wrong |
|  |  |  |
|  |  |  |
|  |  |  |

**Now, write your program in Java.**

**public class volume**

**{**

**public static void main(String args[])**

**{**

**int a=5;**

**double volume=a\*a\*a;**

**double sa= a\*a\*6;**

**System.out.println("Volume of the cube="+volume);**

**System.out.println("Surface Area of the cube="+sa);**

**double r1;**

**double r2;**

**r1=(volume\*3/(4\*3.14));**

**r2=(sa/(4\*3.14));**

**System.out.println("Radius of sphere with same volume(" + volume + ")=" + Math.cbrt(r1));**

**System.out.println("Radius of sphere with same surface area(" + sa + ")=" + Math.sqrt(r2));**

**}**

**}**

**Output:**

**Volume of the cube=125.0**

**Surface Area of the cube=150.0**

**Radius of sphere with same volume(125.0)=3.1022767838176404**

**Radius of sphere with same surface area(150.0)=3.4558175818806856**

B. This problem uses the following Date class. Open Notepad++ and type in or copy the below program code into a blank document and save it as Date.java in the same directory as your program for this problem:

|  |
| --- |
| public class Date {  private int day;  private int month;  private int year;    /\*\* default constructor  \* sets month to 1, day to 1 and year to 2000  \*/  public Date( ) {  setDate( 1, 1, 2000 );  }    /\*\* overloaded constructor  \* @param mm initial value for month  \* @param dd initial value for day  \* @param yyyy initial value for year  \*  \* passes parameters to setDate method  \*/  public Date( int mm, int dd, int yyyy ) {  setDate( mm, dd, yyyy );  }    /\*\* accessor method  \* returns the month  \*/  public int getMonth( ) { return month; }    /\*\* accessor method  \* returns the day  \*/  public int getDay( ) { return day; }    /\*\* accessor method  \* returns the year  \*/  public int getYear( ) { return year; }    /\*\* mutator method \*/  /\*\* setDate  \* @param mm new value for month  \* @param dd new value for day  \* @param yyyy new value for year  \* passes parameters to setMonth, setDay, and setYear for validation and setting  \*/  public void setDate( int mm, int dd, int yyyy ) {  setYear(yyyy);  setMonth( mm );  setDay( dd );  }    /\*\* helper methods \*/  /\*\* setDay (private)  \* @param dd new value for day  \* if dd is legal day for current month, sets day to dd  \* otherwise, sets day to 1  \*/  private void setDay( int dd ) {  int [] validDays = { 0, 31, 29, 31, 30,  31, 30, 31, 31, 30,  31, 30, 31 };  day = ( dd >= 1 && dd <= validDays[month] ? dd : 1 );  }  /\*\* setMonth (private)  \* @param mm new value for month  \* if mm is between 1 and 12, sets month to mm  \* otherwise, sets month to 1  \*/  private void setMonth( int mm ) {  month = ( mm >= 1 && mm <= 12 ? mm : 1 );  }  /\*\* setYear (private)  \* @param yyyy new value for year  \* sets year to yyyy  \*/  private void setYear( int yyyy ) {  year = yyyy;  }    /\*\* toString  \* @return String  \* returns date in mm/dd/yyyy format  \*/  public String toString( ) {  return month + "/" + day + "/" + year;  }    /\*\* equals  \* @param d Date object to compare to this  \* @return true if d is equal to this  \* false, otherwise  \*/  public boolean equals( Date d ) {  if ( month == d.month  && day == d.day  && year == d.year )  return true;  else  return false;  }  } |

Your program should prompt the user for birthday in dd/mm/yyyy format and read it as a String using Scanner (you can assume the user enters the correct format). Parse the String, make a Date object, and output the Date object. Then create a Date object for your (the programmer) birthday and output this message "Your birthday is the same as the programmer birthday (true/false): " and use the Date ``.equals" method to output the boolean answer.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Inputs and outputs (use “N/A”, “undefined”, “none”, etc. If necessary)** | | | | | |
| Variable name | Input or Output? | Data type / format | Constraints | Special cases | Comments |
| Day | Input | INT | dd>=1  dd<=31 | dd<1 |  |
| Month | Input | INT | mm>=1  mm<=12 | mm<1  mm>12 |  |
| Year | Input | INT | yyyy>=1900  yyyy<=2100 | yyyy<1900  yyyy>2100 |  |
| Date | Output | String | Date= dd + mm + yyyy | N/A |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |
| --- |
| **Pseudocode:** |
| GET the day, month, and year.  COMPUTE Date  Date= “dd” + “mm” + “yyyy”  STORE Date  COMPARE Date with programmer’s date  IF Date = programmer’s date, then DISPLAY “ Your birthday is the same as the programmer birthday (true)”  ELSE DISPLAY “Your birthday is the same as the programmer birthday (False)” |

|  |  |  |
| --- | --- | --- |
| **Test case name (ex. “negative height”, “typical conditions”, etc.)** | **Input data set for this test case** | **Explain why you chose this test case** |
| Correct Input | 1<dd<31;1<mm<12; 1900<yyyy<2100 | To test expected results |
| Invalid Day | dd<1 or dd>31 | Check illegal inputs |
| Invalid Month | mm<1 or mm>12 | Check illegal inputs |
| Invalid Year | yyyy<1900 or yyyy>2100 | Check illegal inputs |
|  |  |  |
|  |  |  |

**Java Code and Output (From BlueJ):**

**public class Date {**

**private int day;**

**private int month;**

**private int year;**

**public Date( ) {**

**setDate( 1, 1, 2000 );**

**}**

**public Date( int mm, int dd, int yyyy ) {**

**setDate( mm, dd, yyyy );**

**}**

**public int getMonth( ) { return month; }**

**public int getDay( ) { return day; }**

**public int getYear( ) { return year; }**

**public void setDate( int mm, int dd, int yyyy ) {**

**setYear(yyyy);**

**setMonth( mm );**

**setDay( dd );**

**}**

**private void setDay( int dd ) {**

**int [] validDays = { 0, 31, 29, 31, 30,**

**31, 30, 31, 31, 30,**

**31, 30, 31 };**

**day = ( dd >= 1 && dd <= validDays[month] ? dd : 1 );**

**}**

**private void setMonth( int mm ) {**

**month = ( mm >= 1 && mm <= 12 ? mm : 1 );**

**}**

**private void setYear( int yyyy ) {**

**year = yyyy;**

**}**

**public String toString( ) {**

**return month + "/" + day + "/" + year;**

**}**

**public boolean equals( Date d ) {**

**if ( month == d.month**

**&& day == d.day**

**&& year == d.year )**

**return true;**

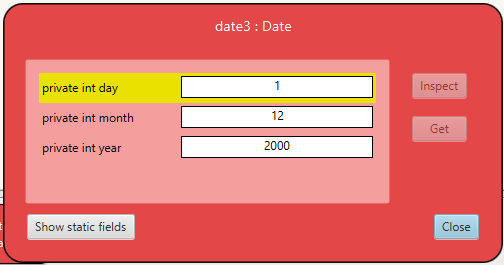
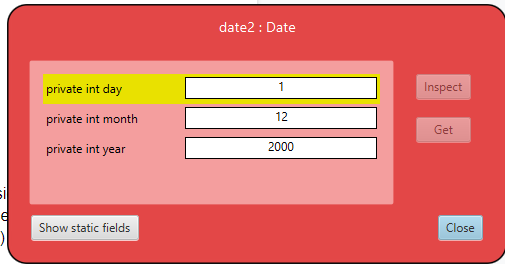
**else**

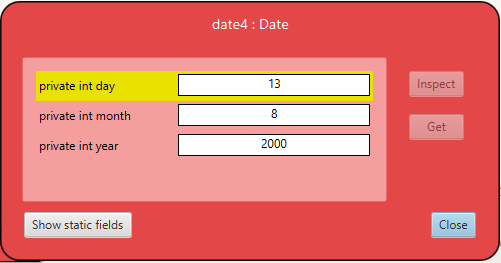
**return false;**

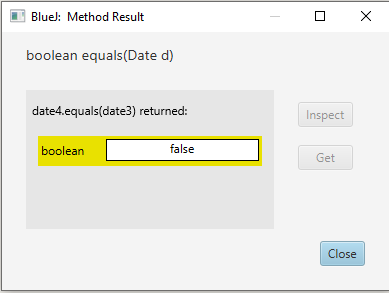
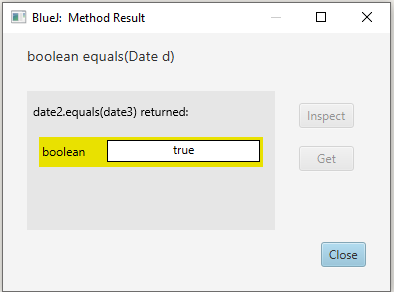
**}**

**}**

**Output**

****

****

****

C. In probability theory, the expected outcome on a roll of fair 6-sided die is 3.5. Write a program to generate 10 rolls of a fair 6-sided die and output the average after 5 rolls and again after all 10 rolls (it should be nearer to 3.5 after 10 rolls) **[5 points]**.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Inputs and outputs (use “N/A”, “undefined”, “none”, etc. If necessary)** | | | | | |
| Variable name | Input or Output? | Data type / format | Constraints | Special cases | Comments |
| Roll 1to Roll 10 | Input | Integer | Rolls>0 |  |  |
| Sum of Rolls from Roll 1 to Roll 5 | Output & Input | Integer | Sum of Rolls>0, Sum of Rolls<30 |  | As it is first computed and output but then used again it is also input |
| Sum of Rolls Roll1 to Roll 10 | Output & Input | Integer | Sum of Rolls>0,  Sum of Height <60 |  | As it is first computed and output but then used again it is also input |
| Average of 5 Rolls | Output | Float | Sum of Rolls>0,  Sum of Height <15 |  |  |
| Average of 10 Rolls | Output | Float | Sum of Rolls>0,  Sum of Height <30 |  |  |
|  |  |  |  |  |  |

|  |
| --- |
| **Pseudocode:** |
| GET Rolls 1 to 10  COMPUTE Sum of Rolls from 1 to 5  Sum of Rolls=Roll1+Roll2+Roll3+Roll4+Roll5  Sum of Rolls from 1 to 10  Sum of Rolls=Roll 1+ Roll2 + Roll 3 + Roll 4 + Roll 5 + Roll 6 + Roll 7+Roll 8+Roll 9+Roll 10  STORE Sum 1 to 5, Sum 1 to 10  COMPUTE Average of Sum 1 to 5, Average of Sum 1 to 10  Average of Sum 1 to 5=(Sum of Rolls 1 to 5) / 5  Average of Sum 1 to 10=(Sum of Rolls 1 to 10) / 10  STORE Average of Sum 1 to 5, Average of Sum 1 to 10  DISPLAY Average of Sum 1 to 5, Average of Sum 1 to 10 |

|  |  |  |
| --- | --- | --- |
| **Test case name (ex. “negative height”, “typical conditions”, etc.)** | **Input data set for this test case** | **Explain why you chose this test case** |
| Control Test | Roll 1= 2,Roll 2=2,Roll 3= 3,Roll 4= 4,Roll 5= 5,Roll 6= 6,Roll 7= 2,Roll 8= 4,Roll 9= 3,Roll 10= 1 | To compare answer to other test cases |
| Odd numbers of Negative Roll | Roll 1= -2,Roll 2=2,Roll 3= 3,Roll 4= 4,Roll 5= 5,Roll 6= 6,Roll 7= 2,Roll 8= 4,Roll 9= 3,Roll 10= 1 | To see if error is shown as roll cannot be negative |
| Even Number of Negative Rolls | Roll 1= -2,Roll 2=-2,Roll 3= 3,Roll 4= 4,Roll 5= 5,Roll 6= 6,Roll 7= 2,Roll 8= 4,Roll 9= 3,Roll 10= 1 | To see if error is shown as even if the answer is correct input can not be negative |
| Roll>6 | Roll 1= 7,Roll 2=2,Roll 3= 3,Roll 4= 8,Roll 5= 5,Roll 6= 6,Roll 7= 2,Roll 8= 4,Roll 9= 3,Roll 10= 1 | To see if error is found as roll cannot be more that 6 |
|  |  |  |
|  |  |  |

**Program in Java and Output:**

**import java.util.Scanner;**

**class roll {**

**public static void main(String args[]){**

**try (Scanner scanner = new Scanner(System.in)) {**

**System.out.println("Please enter the Numbers faced in the 10 rolls (Between 1-6):");**

**double roll1= scanner.nextDouble();**

**double roll2= scanner.nextDouble();**

**double roll3= scanner.nextDouble();**

**double roll4= scanner.nextDouble();**

**double roll5= scanner.nextDouble();**

**double roll6= scanner.nextDouble();**

**double roll7= scanner.nextDouble();**

**double roll8= scanner.nextDouble();**

**double roll9= scanner.nextDouble();**

**double roll10=scanner.nextDouble();**

**double sum1= (roll1+roll2+roll3+roll4+roll5)/5;**

**System.out.println("Average of Sum 1 :"+sum1);**

**double sum2=(roll1+roll2+roll3+roll4+roll5+roll6+roll7+roll8+roll9+roll10)/10;**

**System.out.println("Average of Sum 2 :"+sum2);**

**}**

**}**

**}**

**Output:**

**Please enter the Numbers faced in the 10 rolls (Between 1-6):**

**1**

**2**

**3**

**4**

**5**

**6**

**5**

**4**

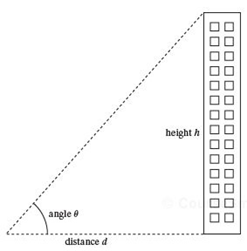
**3**

**2**

**Average of Sum 1 :3.0**

**Average of Sum 2 :3.5**

D. You can use trigonometry to find the height of a building as shown below.



Prompt the user for an estimate on the angle theta in degrees (assume their estimate is within plus or minus 3 degrees) between the line of sight to the top of the building and the ground, and prompt the user for the accurate distance d to the building. Calculate an upper and lower estimate for the height of the building.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Inputs and outputs (use “N/A”, “undefined”, “none”, etc. If necessary)** | | | | | |
| Variable name | Input or Output? | Data type / format | Constraints | Special cases | Comments |
| Theta | input | float | o<t<90 | none | T = 50 |
| Distance | input | float | d>0 | none | D = 70m |
| Upper estimate | input | float | t+3 | none | Ut = 53 |
| Lower estimate | input | float | lt-3 | none | Lt = 47 |
| height | output | Float | h>0 | none | H= = tan(angle)\*distance |
|  |  |  |  |  |  |

|  |
| --- |
| **Pseudocode:** |
| GET Angle of Elevation θ, Distance d  COMPUTE Upper Estimate Hu, Lower Estimate Hl  Hu=tan(θ+3)\*d  Hl=tan(θ-3)\*d  STORE Hu, Hl  DISPLAY Hu,Hl |

|  |  |  |
| --- | --- | --- |
| **Test case name (ex. “negative height”, “typical conditions”, etc.)** | **Input data set for this test case** | **Explain why you chose this test case** |
| Control Test | Distance =10m, Angle=50 degree | To compare the answer to other test as this is the correct answer |
| Negative Distance | d=-10. Angle=50 | To see if it shows error as distance cannot be negative |
| Negative Angle | d=10, Angle=-50 | To see if it shows error as angle cannot be negative |
| Angle>=90 or Angle=0 | d=10, Angle=90 | To see if it shows error as angle cannot be equal or more than 90 |
|  |  |  |
|  |  |  |

**Program in Java and Output:**

**import java.lang.Math;**

**class height {**

**public static void main(String args[])**

**{**

**double a = 30;**

**double b = Math.toRadians(a);**

**double d=10;**

**double hu= d\* Math.tan(b+3);**

**double hl= d\* Math.tan(b-3);**

**System.out.println("The Upper height of building="+hu);**

**System.out.println("The Lower height of building="+hl);**

**}**

**}**

**Output:**

**The Upper height of building=4.017407496568068**

**The Lower height of building=7.844570680937552**