Alan Palayil

Rezwan Rifat

Anwar Adam

**CS 115 Fall 2019 Lab #5**

Due: **Monday, October 7th, midnight**

Points: **20**

**Instructions:**

1. Use this document template to report your answers. Enter all lab partner names at the top of 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\_Lab5\_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. (2 points) Implement selection condition statements.

2. (10 points) Write programs that use selection.

3. (8 points) Write and test user-defined class.

**Problem 1 [2 points]:**

For these problems you do not need to write entire Java program, just the condition statements requested. You can assume that all referenced variables have already been DECLARED and INITIALIZED.

A. **[0.5 points]** Write statements containing a logical expression that assigns true to the boolean variable isCandidate if satScore is greater than or equal to 1100, gpa is not less than 2.5, and age is greater than 15. Otherwise, isCandidate should be false.

|  |
| --- |
| **Your answer:** |
| if (satScore >=1100) && (gpa>=2.5)&&(age>=15){  isCandidate=true  }  else {  is Candidate=false  } |

B. **[0.5 points]** Write a selection statement that will print "hot" if the temperature >= 80, "pleasant" if the temperature >= 60, "cool" if the temperature is >= 45, "cold" otherwise.

|  |
| --- |
| **Your answer:** |
| If ( temperature>=80){  System.out.println( “hot”);  }  Else if ( temperature>=60) && (temperature<80){  System.out.println(“pleasant”);  }  Else if (temperature>=40) && (temperature<60){  System.out.println(“cool”);  }  Else{  System.out.println(“cold”);  } |

C. **[0.5 points]** If integer variable currentNumber is odd, change its value so that it is now 3 times currentNumber, then add 1, otherwise change the sign of currentNumber.

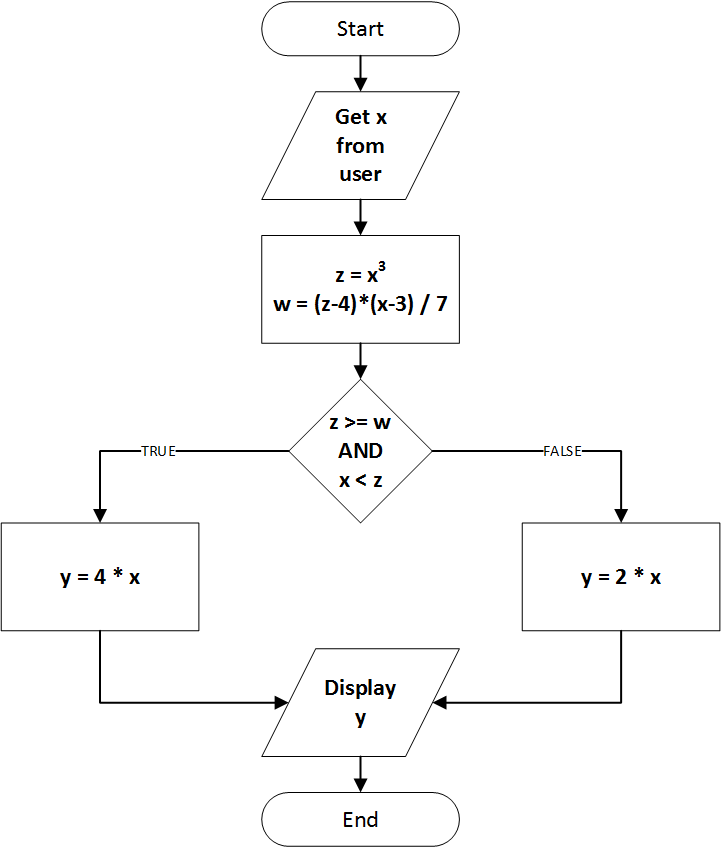
|  |
| --- |
| **Your answer:** |
| If (currentNumber%2==1){  currentNumber=currentNumber\*3+1;  }  Else if (currentNumber%2==0){  currentNumber=currentNumber\*-1;  } |

D. **[0.5 points]** Find the largest of three numbers x, y, and z and assign its value to the variable largest.

|  |
| --- |
| **Your answer:** |
| If (x>y) && (x>z){  largest=x;  }  Else if ( y>x)&& (y>z){  largest=y;  }  Else if (z>x)&& (z>y){  largest=z;  } |

**Problem 2 [5 points]:**

Implement the following program flowchart in Java.



You can use the skeleton code provided below.

|  |
| --- |
| **Your Java program:** |
| import java.util.Scanner;    public class Lab5Problem2 {    public static void main(String args[]){  // Your code goes here…  double x;  double z;  double w;  double y;  Scanner scan = new Scanner(System.in);  System.out.println("Enter x=");  x = scan.nextDouble();  z=Math.pow(x,3.0);  w=(z-4)\*(x-3)/7;    if ((z>=w) && (x<z)){  y=4\*x;  }  else {  y=2\*x;  }  System.out.println(y);      }  } |

Here are some sample program outcomes.

|  |
| --- |
| **Sample behavior:** |
| Give me x: 2.0  The value of y is: 8.0    Give me x: 0.2  The value of y is: 0.4    Give me x: -3.0  The value of y is: -6.0 |

**Problem 3 [5 points]:**

Recall the following problem from the beginning of the course:

*Given the 3 dimensions of a box (length, width, and height), multiply them together to determine the volume.*

Your task is to write a program (similar to that from Problem 4) that:

n Ask the user for input: length, width, and height,

n Check if all three inputs are valid and can be used:

n If all three are valid, display a message on screen saying: “The volume is: 3.56” (assuming this is the value that was calculated),

n If they’re not, set the volume to zero and display a message on screen: “ERROR: Cannot calculate volume.”

You can use the skeleton code provided below.

|  |
| --- |
| **Your Java program:** |
| import java.util.Scanner;  public class Lab5Problem3  {  public static void main(String args[])  {  try(Scanner scanner= new Scanner(System.in))  {  System.out.println(" Please enter the length, width, and height of the box");  double length= scanner.nextDouble();  double width= scanner.nextDouble();  double height=scanner.nextDouble();  if(length>0 && width>0 && height>0)  {  double volume= length\*width\*height;  System.out.println("The volume of the box is:" +volume);  }  else  {  System.out.println("ERROR: Cannot calculate volume");  System.out.println("The volume of the box is: 0");  }  }  }  }  /\*Output-  NPP\_EXEC: "Run Java"  CD: D:\Notepad++  Current directory: D:\Notepad++  "C:\Program Files\Java\jdk1.8.0\_202\bin\java" Lab5Problem3  Process started (PID=6420) >>>  Please enter the length, width, and height of the box  5  6  7  The volume of the box is:210.0  <<< Process finished (PID=6420). (Exit code 0)  ================ READY ================    \*/ |

**Problem 4 [8 points]:**

Write a user-defined Java class called BoxVolumeCalculator that could be used to calculate the volume of a box given length, height, and width (similar to your program from Problem 3).

Your class should:

n NOT have a main method,

n Should have four fields (use correct data types):

n length,

n width,

n height, and

n volume,

n A no-argument constructor method that resets all four fields to zero,

n Three mutator / setter methods:

n setLength( ) that will set the length field to a new value passed as argument (regardless of whether it is valid or not),

n setWidth( ) that will set the width field to a new value passed as argument (regardless of whether it is valid or not),

n setHeight( ) that will set the height field to a new value passed as argument (regardless of whether it is valid or not),

n Use public access modifiers for all four mutator / setter methods,

n Four accessor / getter methods:

n getLength() that will return the value of the length field,

n getWidth() that will return the value of the width field,

n getHeight() that will return the value of the height field,

n getVolume() that will return the value of the volume field,

n Use public access modifiers for all four accessor/ getter methods,

n a calculateVolume() method which:

n If all three dimension fields have valid values will calculate box volume based on current values of length, height, and width fields and update the volume field accordingly.

n Otherwise it will set the volume field to zero

If you implemented your BoxVolumeCalculator class correctly and compile/run the following program:

|  |
| --- |
| Lab5Problem4Test **program:** |
| public class Lab5Problem4Test {    public static void main(String args[]){  // Instantiate the BoxVolumeCalculator class object (calling constructor)  BoxVolumeCalculator myCalculator = new BoxVolumeCalculator();    // Show field values before using setter methods to change them  System.out.println("Initial dimensions:");  System.out.println("Length: " + myCalculator.getLength() + " | Width: " + myCalculator.getWidth() + " | Height: " + myCalculator.getHeight());    // Set box dimensions using mutator / setter methods  myCalculator.setLength(2.0);  myCalculator.setWidth(3.0);  myCalculator.setHeight(4.0);    // Show field values before using setter methods to change them  System.out.println("New dimensions:");  System.out.println("Length: " + myCalculator.getLength() + " | Width: " + myCalculator.getWidth() + " | Height: " + myCalculator.getHeight());    // Re-calculate box volume  myCalculator.calculateVolume();    // Show the current box volume  System.out.println("Calculated volume: " + myCalculator.getVolume());    // Let's try again with invalid dimensions  System.out.println("Let's try again:");    // Reset box dimensions using mutator / setter methods  myCalculator.setLength(2.0);  myCalculator.setWidth(3.0);  myCalculator.setHeight(-4.0);    // Show field values before using setter methods to change them  System.out.println("New, reset, dimensions:");  System.out.println("Length: " + myCalculator.getLength() + " | Width: " + myCalculator.getWidth() + " | Height: " + myCalculator.getHeight());    // Re-calculate box volume  myCalculator.calculateVolume();    // Show the current box volume  System.out.println("Calculated volume: " + myCalculator.getVolume());      }  } |

You should see the following output:

Initial dimensions:

Length: 0.0 | Width: 0.0 | Height: 0.0

New dimensions:

Length: 2.0 | Width: 3.0 | Height: 4.0

Calculated volume: 24.0

Let's try again:

New, reset, dimensions:

Length: 2.0 | Width: 3.0 | Height: -4.0

Calculated volume: 0.0

In the program above, note how the calculator object is instantiated and how its methods are being used to achieve the goal.

Our BoxVolumeCalculator class design is not the most practical. Can you identify any flaws? What could go wrong and what would you change (you don’t need to modify code, just explain it):

|  |
| --- |
| **Your answer:** |
| The program runs fine, the only change I would suggest is it should calculate the volume as soon as any value length, width, or height is changed.  The code is:  public class BoxVolumeCalculator{  private double length, width, height, volume;    public BoxVolumeCalculator(){  length = 0.0;  width = 0.0;  height = 0.0;  volume = 0.0;  }  public BoxVolumeCalculator(double length, double width, double height){  this.length = length;  this.width = width;  this.height = height;  calculateVolume();  }    public void setLength(double length){  this.length = length;  }  public void setWidth(double width){  this.width = width;  }  public void setHeight(double height){  this.height = height;  }    public double getLength(){  return length;  }  public double getWidth(){  return width;  }  public double getHeight(){  return height;  }  public double getVolume(){  calculateVolume(); // Change I believe is required  return volume;  }    public void calculateVolume(){  if( (length>0)&&(width>0)&&(height>0) ){  volume = length\*width\*height;  System.out.println("The volume is:" +volume);  }  else{  volume = 0;  System.out.println("The volume is:" +volume);  System.out.println("The values are invalid!");  }  }  }  public class Lab5Problem4Test {    public static void main(String args[]){  // Instantiate the BoxVolumeCalculator class object (calling constructor)  BoxVolumeCalculator myCalculator = new BoxVolumeCalculator();    // Show field values before using setter methods to change them  System.out.println("Initial dimensions:");  System.out.println("Length: " + myCalculator.getLength() + " | Width: " + myCalculator.getWidth() + " | Height: " + myCalculator.getHeight());    // Set box dimensions using mutator / setter methods  myCalculator.setLength(2.0);  myCalculator.setWidth(3.0);  myCalculator.setHeight(4.0);    // Show field values before using setter methods to change them  System.out.println("New dimensions:");  System.out.println("Length: " + myCalculator.getLength() + " | Width: " + myCalculator.getWidth() + " | Height: " + myCalculator.getHeight());    // Re-calculate box volume  myCalculator.calculateVolume();    // Show the current box volume  System.out.println("Calculated volume: " + myCalculator.getVolume());    // Let's try again with invalid dimensions  System.out.println("Let's try again:");    // Reset box dimensions using mutator / setter methods  myCalculator.setLength(2.0);  myCalculator.setWidth(3.0);  myCalculator.setHeight(-4.0);    // Show field values before using setter methods to change them  System.out.println("New, reset, dimensions:");  System.out.println("Length: " + myCalculator.getLength() + " | Width: " + myCalculator.getWidth() + " | Height: " + myCalculator.getHeight());    // Re-calculate box volume  myCalculator.calculateVolume();    // Show the current box volume  System.out.println("Calculated volume: " + myCalculator.getVolume());      }  } |