Rochester Institute of Technology
National Technical Institute for the Deaf
Information and Computing Studies Department

Name: Edward Riley

NACA.161 Programming Fundamentals II
In-class Exercise Day 07 – Decisions

Overview

To learn how to get your programs to make decisions.

Create a Class

1) Create a class called **MyMath** that will eventually contain the methods you will create in this exercise. Save and compile the file.

   Create a second class called **TestMyMath** that you will use to test the methods that you will create in the **MyMath** class.

   Save and compile the file.

Create a method

2) Create a method called **absValue** in **MyMath** that will:
   - Accept an integer number
   - Return the absolute value of the number

   HINT: you need to check the number's value. If the number is greater than zero then return the number, but if the number is less than zero then return the number without the negative sign.

   How do you remove the negative sign?

   Multiply it by -1

   Save and compile the file.

Create a Test Class

3) Now you need to see if your code worked. What do you need to crate in your test class in order to call the method you just created?

   (a. absValue(Number))

4) Go ahead and create the code needed to call the **absValue** method. For now, call the method without passing arguments to it.

5) Compile the code. Why did it fail to compile?

Because you need to pass argument

6) In order for the test class to be as versatile as possible, where should it get the number to pass to the method?

from MyMathTest

What class do you use to get user input?

myMath

What else should you do before calling a **Scanner** method?

import java.util.Scanner;

7) Add code to get a number from the user and call the **absValue** method.

8) Compile and run the test class.

9) Why didn't you get any output?

Because I did not put it in System.out.println();

How do you get the value that the **absValue** calculated?

You put it in return and put in system.out.print on MyMathTest im. MyMath

10) Add the code to get the returned value from the method and print the result to the screen.

11) Compile and run the program several times with positive and negative numbers to make sure your code in the **absValue** is working.

Adding Another Method

12) Create another method called **power** that accepts two numbers. The method will take the first number and raise it to the power given in the second number.

For example:

- If the first number is 8 and the second number is 2, your code should raise 8 to the second power (which is 8^2).
- If the first number is 10 and the second number is 3, your code should raise 10 to the third power (which is 10^3).
- If the first number is 13 and the second number is 4, your code should raise 13 to the fourth power (which is 13^4).

To make the coding easier, only use values of 2, 3, and 4 for the second number.

HINT: Since there isn't a Java operator that raises a number to a specified power, just multiply the number by itself the appropriate number of times.

13) Add the required code in the test class that will call the method with 2 numbers from the user and print the returned value.

14) Compile and run the program several times to make sure it works.

One more method

15) Create another method in the **MyMath** class that indicates if a number is even or odd. Make the method return true if it is even and return false if it is odd.

What is the return type of the method?

int

What operator did you use in your if-statement to make this work?

==

16) Add the required code in the test class to test the method you just created in the previous step.

17) Compile and test the program a few times to make sure it all works.

When you complete all of the steps successfully and answer all of the questions, contact your instructor to check if your program(s) executes correctly and to review your code. We will initial the line below.

_____Successful execution of code

If you do not finish the program during the class period, contact your instructor to initial below so that you can complete it before the next class period.

_____Code not completed during lab time

You may then have your instructor verify your work at the start of work period in the next class. If you do not have a signature, then you can not receive any points for this assignment.