

Java Programming for Non-Specialists

CGS 3416: Spring 2021
Department of Computer Science, Florida State University

Homework 2: 100 points

Due date: 11:59 PM 02/17/2021

1 Objective

The objective for this assignment is to familiarize yourself with loops and selection statements in Java. For this assignment, you're required to turn in three programs that involve selection, loops on different levels, and methods.

Please submit your files "Primes.java" and "Powers.java" and "Evaluator.java" to Canvas.

2 Program 1

This program is called "Powers.java".

Write a program to print all the powers of 2 below a certain number and calculate their sum. Make sure your program conforms to the following requirements:

1. Accept the upper limit from the user (as an integer). (5 points)
2. These numbers grow very fast. Make sure the sum variable is of the "long" type. You can assume that the test output will be less than LONG MAX. (5 points)
3. Print all the numbers as a running sum, and finally print their sum. The input and output should match the sample run perfectly. (15 points)

2.1 Sample Runs

There are 2 sample runs here.

Enter the upper limit: 500
1+2+4+8+16+32+64+128+256=511

Enter the upper limit: 1055
1+2+4+8+16+32+64+128+256+512+1024=2047

3 Program 2

This program is called "Primes.java".

Write a program to print all the prime numbers below a certain given number. A prime number is defined as a number that can only be divided by 1 and itself. For example, 2, 3, 17 and 41 are all prime numbers. Make sure your program conforms to the following requirements:

1. Accept the upper limit from the user (as an integer). (5 points)
2. You can assume the user input will be positive and smaller than INT MAX.
3. Go from 2 to the number. If you happen to find a number that is prime, print it. The input and output should match the sample run perfectly. (20 points)

3.1 Sample Runs

There are 2 sample runs here.

Enter the upper limit: 25

The prime numbers are:

2, 3, 5, 7, 11, 13, 17, 19, 23

Enter the upper limit: 70

The prime numbers are:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67

4 Program 3

This program is called "Evaluator.java".

Write a program to evaluate a mathematical expression as it is entered by the user. Here, the user will enter alternating numbers and operators, terminating with a '.'. You are required to keep a running evaluated result for the expression and finally print the output when the user is done. Each 'type' of operation needs to have its own method. The expression will only consist of integers and you only need to account for the 5 basic integer math operations (+, -, *, / and %). Make sure your program conforms to the following requirements:

- Write 5 methods, one for each basic arithmetic operators. They take 2 parameters each and return a result. (25 points)
- In the main method, read a number and a character alternatively until you encounter a '.'. (10 points)
- Keep a running result variable, where you store the result of the calculation so far. (10 points)
- Once the user is done entering the expression, print out the result. (5 points)
- You can assume all the numbers are integer.
- Make sure to call the functions to do the calculations. Don't do the calculations in main().
- You can assume that the input will be as expected. You don't need to do any error checking here.
- You can ignore operator precedence. Just evaluate expressions as they come in.

4.1 Sample Run

There are 2 sample runs here:

Enter the expression:

1
+
26
*
2
%
19
/
5

.
The result is 3

Enter the expression:

12
*
9
-
37
+
-8
.

The result is 63

5 Generic Grading Guidelines

- Please make sure you're only using the concepts already discussed in class. That is, please try and restrict yourself to loops, selection statements, calls to library methods, and your own written methods.
- The first program is worth 25 points, the second is worth 25, and the third program is worth 50 points.
- You do not need to perform any form of error checks.
- Please make sure that you're conforming to specifications (program name, print statements, expected inputs and outputs etc.).
- Please make sure your code is readable.
- Please make sure you've compiled and run your program before you turn it in. If your program does not compile, a grade 0 will be awarded.