CMPU 102 Lab 2: Using iterative and decision statements

Program 1:

1. Log in to your Vassar CS account.

2. Open the cs102 directory you created in the last lab by

entering the "change directory" command as follows:

cd cs102

3. Change directory into the labs subdirectory by entering

cd labs

4. Create a new directory for lab 2 by entering:

mkdir lab2

5. Open the directory you created in the last step by entering:

cd lab2

For each of the 3 classes listed below, create a main

method and write the executable code in the main method.

Program 1: FindSmallest.java

a. Prompt for and read positive integers from the keyboard

until the user enters 0. Use the java.util.Scanner class

to take in input using the nextInt() method. (Be nice to

the user and say please in your prompt.)

b. You need to print each number as entered and store

the smallest positive integer entered so far in a variable.

The initial values of the int entered and the smallest

int should initially be set to the MAX\_VALUE constant of

the Integer class so the values entered by the user must

be smaller.

c. After the user enters 0, the input should stop and you

should print the smallest number entered to the screen.

Your program should work correctly if any number of

integers, even some with equal value, are entered.

\*\*Expected screen output from running this program\*\*:

Please enter any number of positive integers and I will

find the smallest. Enter 0 to stop reading numbers.

4

9

18

22

5

0

The smallest value is 4.

or

Please enter any number of positive integers and I will

find the smallest. Enter 0 to stop reading numbers.

43 75 8 9 7 67 0

The smallest value is 7.

Note that the numbers can be entered one per line or all on

one line, separated by spaces.

Program 2: AssignGrade.java

a. Create a class called "AssignGrade.java" in the lab2

directory. This class should read a score from the

user and calculate a character grade from the score.

b. Prompt for and read a score (a possible real number) from

the keyboard using the showInputDialog of the JOptionPane

class.

c. Use the following scoring policy:

score >= 90 A

80 <= score < 90 B

70 <= score < 80 C

60 <= score < 70 D

score < 60 F

d. Use a series of "if", " if else" and "else" statements to

assign a letter grade of type char in each branch.

e. Once you have assigned a letter grade, use a switch state-

ment (your instructor will explain how a switch statement

works at the start of the lab) to print the letter grade

along with one of the following comments:

Letter grade Comment

------------ -------

'A' Outstanding work!

'B' Very good work!

'C' Nice work.

'D' Good try.

'F' Happy trails.

Look up the switch statement in the Java tutorial if you are not

sure how to use it.

\*\*Expected screen output from running this program\*\*:

Please enter a score between 0 and 100: 98.5

The grade is A: Outstanding work!

Test your program a few times with different scores to convince

yourself it works.

Program 3: AssignGradeCheck.java

After you save the AssignGrade program, change the name of the

AssignGrade program to AssignGradeCheck.java. Be sure to change

the class name of the AssignGrade class to AssignGradeCheck too.

This program should do what the AssignGrade program did, except

for the following added feature in the main method:

a) Check that the user does actually enter a number between

0.0 and 100.0.

After you have read a score, add a while statement that

tests the score to see if it is out of range, and, if the

score entered is out of range, displays a message prompting

for a number between 0.0 and 100.0 and reads the score

again. This loop should continue until a number in the correct

range is entered. Look up the do while loop for this part.

b) The rest of this program should be just like the AssignGrade

program.

\*\*Expected screen output from running this program\*\*:

Please enter a score between 0 and 100: 998

That score is out of range.

Please enter a score between 0 and 100: 79

The grade is C: Nice work.

Test your program a few times to convince yourself it works.

BE SURE TO HAVE A COACH OR YOUR PROFESSOR SEE YOUR WORKING CODE AND

HAVE THEM CHECK YOUR NAME OFF ON THE LAB ROSTER AS HAVING COMPLETED

THE LAB. IF YOU KEEP THE LAB IN YOUR CS ACCOUNT, YOU CAN ALWAYS

PRODUCE IT IF ASKED (AS MIGHT BE THE CASE IF SOMEHOW YOU DID NOT GET

CHECKED OFF FOR THE LAB).

COMPRESS THE lab2 DIRECTORY AND SUBMIT IT ON THE COURSE MOODLE PAGE.