

## 60-140 INTRODUCTION TO ALGORITHMS AND PROGRAMMING I

Dr. Ziad Kobti

### Lab 2: Introduction to decision making and relational operators

#### Objectives:

1. Using basic arithmetic operators in C.
2. Using IF-ELSE statement along with relational operators.

#### Instructions:

Recall in the previous lab you learned how to create a directory, change into it, and view its contents. (i.e. the UNIX commands you need to understand here are `mkdir directory_name`, `cd directory_name`, and `ls`)

In this lab you are asked to create 4 simple but complete C programs, one for each question listed below, in a directory called **lab2** which should be located inside the **cs140** directory you have already created in the last lab. The reason for this is to keep your files organized.

Once you setup your directory structure you can start writing the separate C programs for the following problems (name your files using this naming convention: **lab2\_qn.c** to be consistent, where *n* is the question number (e.g. lab2\_q1.c, lab2\_q2.c, etc.). The code for each program will be kept in a separate file and fully documented. Hint: create a template blank C file to always work from.

At the end of this lab class, you will have 4 source code files saved in a folder called lab2 under the cs140 folder. Recall that for the purpose of evaluating and grading your work by the GA's, you would need to create one script file which holds the source code, compilation result and the output of all programs. The process of creating such a file is described below:

```
>> script lab2.txt
>> cat lab2_q1.c      // will show and add the source code for the first question to
                      // the script file
>>cc lab2_q1.c        // will compile lab2_q1.c
>>./a.out             // will run the program and appends its output to lab2.txt file
>>cat lab2_q2.c      //repeating the process for the second question
>>cc lab2_q2.c
>>./a.out
>>cat lab2_q3.c      //repeating the process for the third question
>>cc lab2_q3.c
>>./a.out
>>cat lab2_q4.c      //repeating the process for the fourth question
>>cc lab2_q4.c
>>./a.out

>>exit               //closes the script file. Your work is ready to be graded!
```

#### Submission:

Your lab is graded either at the end of your current lab class (or at the very beginning of your next regular lab class without penalty). Late labs without a valid excuse (eg. illness) receive 0. You are to present your code on the computer in the lab to your lab instructor to receive a grade as follows:

0= not satisfactory or no documentation; 1=incomplete; 2=complete and well documented

**Questions (each to be coded in a separate C file):**

**Lab2\_q1.c**

**Write a program that reads in five integers and then determines and prints the largest and the smallest integers in the group. Use only the programming techniques learned so far. (Do not use arrays!)**

Example:

```
Please enter five integers:  5 12 -10 0 1
The Max is : 12
The Min is : -10
```

**Lab2\_q2.c**

**Write a program that reads an integer and determines and prints whether it is odd or even (Hint: use the remainder operator)**

Example:

```
Please enter an integer:  5
5 is odd!
```

**Lab2\_q3.c**

**Write a program that reads in two integers and determines and prints if the first is a multiple of the second (Hint: use the remainder operator)**

Example:

```
Please enter two integer:  15 5
15 is a multiple of 5

Please enter two integer:  11 5
11 is not a multiple of 5
```

**lab2\_q4.c**

**Write a program that inputs one five-digit number, separates the number into its individual digits and prints the digits separated from one another by three spaces each. (Hint: you need to use combinations of integer division and remainder operators)**

Example:

```
Please enter a five digit number:  15389
The digits of 15389 are: 1   5   3   8   9
```

**BONUS + 1 point (can be received only if you complete all 4 questions correctly)**

**lab2\_q5.c**

Question lab2\_q4 is designed to run once. That is the program sequence is to prompt the user for a number, input the number, then compute and output the digits. Now what if we want the same program to prompt the user if he/she likes to enter another number, if so, the program logic repeats, prompting for the new number and displaying its digits. Of course, the program would repeat until the user answers no, he/she does not wish to enter another number. See example below:

```
Please enter a five digit number:  15389
The digits of 15389 are: 1   5   3   8   9
Would you like to try again? Y
Please enter a five digit number:  20010
The digits of 15389 are: 2   0   0   1   0
Would you like to try again? N
Goodbye!
```