

Assignment 2 – Control Statements/User-Defined Functions

Deadline: Monday July 17 at 23:59
Type: Individual Assignment
Weight: 4%

Submission instructions:

- Create a cpp file for each question
- Compress the files using zip or other tools
- Submit the zip file on Moodle
- Please do not submit exe files
- All submissions must be done through Moodle

Marking Scheme:

- Program correctness (90%)
 - Program clarity (output format, comments, completeness, readability) (10%)
-

Q1. (50 marks) You are asked to write a C++ program which draws a house with a roof based on the following specifications.

Application name: Display a welcome banner

- 1) *Welcome user:* Ask the user for their name and using their name welcome them to your application.
- 2) *Request house dimensions and validate input:* Ask the user to enter the width and height of the house to be drawn (Note: Both height and width are integer). The width must be even and bigger than 2. If the user enters odd numbers or a number less than or equal to 2 for the width, you are required to prompt the user until they enter an even number. They have 3 tries for entering width. If after 3 tries they are still entering odd numbers terminate your program with an appropriate personalized message otherwise move on to step 3.
- 3) *Draw the house*
 - a. *Draw the roof:*
 - i. The roof consists of a set of stars on each row. Number of stars in the last row of the roof is equal to the width of the house. The first row starts with two stars and you increase the number of starts in the next row by 2 and repeat this process until you reach to the width. For example if the width is 6, the roof shape will be like this (2,4 and 6 stars):

**

Hint: The number of rows needed to print/draw the roof is half the width of the house.

Note: There is no space between the stars in each row.

b. Draw the body of the house:

- i. The body of the house has *height+1* rows in all.
- ii. Last row are drawn using the dash character (-). There are *width* dashes.
- iii. The walls are represented by *height* rows. Each of the rows are made up of 2 characters of | in the left and right sides and the rest are spaces.

c. Keep track of the number of houses you have drawn.

- 4) *Again?* Ask the user if they wish you to draw another house. If yes repeat steps 3. If no, move on to step 5.
- 5) *End program:* display this message: "Hope you like your house(s)"

Here are a few sample outputs: user input is highlighted in grey

```
-----  
House Drawing Program  
-----  
  
What is your name? Anna  
Well Anna, welcome to the house drawing program.  
Do you want me to draw a simple house for you? (yes/no) yes  
  
Enter height of the house you want me to draw: 3  
Please enter an even number for the width of the house (must be even  
numbers and bigger than 2): 3  
You enter 3 for the width. Not an even number!  
  
Please enter an even number for the width of the house (must be even  
numbers and bigger than 2): 5  
You enter 5 for the width. Not an even number!  
  
Please enter an even number for the width of the house (must be even  
numbers and bigger than 1): 11  
  
You enter 11 for the width. Not an even number!  
  
it seems you are having troubles entering even numbers! Program ends now.
```

House Drawing Program

What is your name? Anna

Well Anna, welcome to my silly house drawing program.

Do you want me to draw a simple house for you? (yes/no) yes

Enter height of the house you want me to draw: 3

Please enter an even number for the width of the house (must be even numbers and bigger than 2): 6

* * * *

||

Do you want me to draw a simple house for you? (yes/no) **yes**

Enter height of the house you want me to draw: 5

Please enter an even number for the width of the house (must be even numbers and bigger than 2): 10

* * * * *

Do you want me to draw a simple house for you? (yes/no) no

Hope you like your 2 houses!

Q2. (50 marks) Write a C++ program that asks the user to enter two positive integer numbers as the lower bound and upper bound. Then it asks the user to enter a character:

- If the entered character is 'a', **function1** is called.
- If the entered character is 'b', **function2** is called and then the value of **result** variable is printed
- If the entered character is 'c', **function 3** is called and the returned value of the function 3 is printed.
- If the user enters any other character, the program prints "invalid input" and terminates.

Function 1:

This function accepts the upper bound and lower bound numbers as the input arguments and prints out all the numbers in this range (Inclusive) which are multiples of both 3 and 7.

Function 2:

This function has no return value. It accepts 3 input arguments: the **upper bound** and **lower bound** numbers and a variable **result** (by reference) and calculates the difference between two entered numbers and save it in the **result**.

Function 3:

This function returns a variable of type double (**sum**) and accepts the upper bound and lower bound numbers (*lower* and *upper* variables) as input arguments. It calculates the results of following equation and returns the **sum** variable. (please note that the number of digits after the decimal point should be set to 3 for the **sum** value).

$$sum = \frac{1}{lower} + \frac{1}{lower+1} + \frac{1}{lower+2} + \dots + \frac{1}{upper}$$

Here are several sample outputs:

```
Please enter two positive integer numbers: (Lower bound/Upper bound): 11 63
Please enter a character: a
List of numbers in this interval which are multiple of both 3 and 7: 21 42 63
```

Please enter two positive integer numbers: (Lower bound/Upper bound): 11 63

Please enter a character: b

The difference between two numbers is 52

Please enter two positive integer numbers: (Lower bound/Upper bound): 20 25

Please enter a character: c

the value of sum is: 0.268

Please enter two positive integer numbers: (Lower bound/Upper bound): 20 25

Please enter a character: z

Invalid input