Programming Fundamentals – Spring 2013

(BS-SE-F12 Morning & Afternoon)

Assignment # 2

Submission Deadline: **Tuesday, 23rd April, 2013 (till 6:00 PM)**

Submission Folders:

\\printsrv\Teacher Data\Ahmad Ghazali\PF - S13\A2 Morning

\\printsrv\Teacher Data\Ahmad Ghazali\PF - S13\A2 Afternoon

Instructions

- **This is an individual assignment. You are NOT allowed to work/submit in form of group. Absolutely NO collaboration is allowed. Any traces of plagiarism/cheating would result in an "F" grade in this course.**
- Do **NOT** copy even a single line of code from any other person or book or Internet or any other source.
- This assignment needs to be submitted in **Soft Copy**. See **Submission Procedure** at the end.
- Late submissions will NOT be accepted, in any case.
- Clearly mention your **Name**, **Roll Number** and **Section** in comments at the top of each CPP file.

In this assignment you are required to solve and submit C++ programs for the following 5 problems:

1. Write a **C++ program** which inputs two integers from user and display the following sequence in the specified format. [**Hint:** Use nested loops]

   Sample Runs are given **for text messages** to be used for input and output. (Input is underlined in sample run, only to distinguish from display messages)

   Sample Run 1:

```
Enter starting integer: 1
Enter ending integer: 5
(1,1)(1,2)(1,3)(1,4)(1,5)
(2,2)(2,3)(2,4)
(3,3)
```

   Sample Run 2:

```
Enter starting integer: 11
Enter ending integer: 18
(11,11)(11,12)(11,13)(11,14)(11,15)(11,16)(11,17)(11,18)
(12,12)(12,13)(12,14)(12,15)(12,16)(12,17)
(13,13)(13,14)(13,15)(13,16)
(14,14)(14,15)
```

2. A right triangle can have sides that are all integers. A set of three integer values for the sides of a right triangle is called a *Pythagorean triple*. These three sides must satisfy the relationship that the sum of the squares of two of the sides is equal to the square of the hypotenuse. Find and display all Pythagorean triples for side1, side2 and hypotenuse all no larger than 200. (Hint: Use a **triple-nested for loop** that tries all possibilities.)

3. Write a C++ program which takes a positive integer *N* from the user, and prints the multiplication table of size *N* by *N*. For example, if the user enters **10**, the following table should be displayed by your program.

```
      1   2   3   4   5   6   7   8   9  10
   +----------------------------------------
 1 |   1   2   3   4   5   6   7   8   9  10
 2 |   2   4   6   8  10  12  14  16  18  20
 3 |   3   6   9  12  15  18  21  24  27  30
 4 |   4   8  12  16  20  24  28  32  36  40
 5 |   5  10  15  20  25  30  35  40  45  50
 6 |   6  12  18  24  30  36  42  48  54  60
 7 |   7  14  21  28  35  42  49  56  63  70
 8 |   8  16  24  32  40  48  56  64  72  80
 9 |   9  18  27  36  45  54  63  72  81  90
10 |  10  20  30  40  50  60  70  80  90 100
```

Note: The output of your program should follow the above output format exactly. You will have to use appropriate format manipulator(s) to achieve this.

4. Write a C++ program which prints the following pattern by taking the height and width from the user. For example, if the user enters height to be **5** and width to be **11**, the following pattern should be displayed on screen:

```
++++++++++
+ + + + + +
++++++++++
+ + + + + +
++++++++++
```

In order to display the above pattern, your program should use ONLY following THREE cout statements:

```
cout << "+";   cout << " ";   cout << endl;
```

*Input validation: Your program should make sure that both height and width are **odd numbers**. Moreover, the value of **height should be at least 3** and value of **width should be at least 5**. In case of invalid input, your program should keep prompting the user again and again till the user provides valid input.*

5. Write a C++ program which prints the following pattern by taking the size (dimensions) from the user. For example, if the user enters the size to be **5**, the following pattern should be displayed on screen:

```
*****
 *****
  *****
   *****
    *****
```

In order to display the above pattern, your program should use ONLY following THREE cout statements:

```
cout << "*";   cout << " ";   cout << endl;
```

*Input validation: Your program should make sure that the size is **greater than or equal to 2**. In case of invalid input, your program should keep prompting the user again and again till the user provides valid input.*

Submission Procedure

You are required to submit this assignment in soft copy format (in the specified folder). Put the **five .CPP files (q1.cpp, q2.cpp, …, q5.cpp)** in a folder (do NOT include any other files in your submission). The name of the folder should be your complete roll number (like BSEF12M234), then **compress** that folder, and copy the **.RAR** file in the specified submission folder for your section.

Please note:

- These *good programming practices* will also have their (significant) weightage in the marking of this assignment:
  - Comment your code intelligently. **Uncommented code *will* not be given any credit.**
  - Indent your code properly.
  - Use meaningful variable names.
  - Use meaningful prompt lines/labels for input/output.
- Although, there are five questions in this assignment, but only two or three questions will be marked. Since, you do not know which those two/three questions are, therefore, you are advised to solve and submit **all five questions** ☺.

☺ GOOD LUCK! ☺

Remember: Honesty always gives fruit (no matter how frightening is the consequence); and Dishonesty is always harmful (no matter how helping it may seem in a certain situation)!