



Introduction to Programming Language (ITP101)

Assignment 1

Units of Focus: { unit 1, unit 2, unit 3 }

GENERAL INSTRUCTIONS

- THIS IS AN INDIVIDUAL ASSIGNMENT. READ THE GIVEN QUESTIONS CAREFULLY AND PREPARE YOUR ANSWERS ACCORDINGLY.
- YOU MUST UPLOAD A ZIP OF YOUR ANSWERS ON VLE. FOLLOW THE FOLLOWING FILE NAMING CONVENTION:
`REGISTRATIONNUMBER_ASSIGNMENT1.ZIP`.
- ALL YOUR FLOWCHARTS MUST BE DESIGNED USING RAPTOR FOR THIS ASSIGNMENT. NO HAND-DRAWN FLOWCHART SHALL BE ACCEPTED. USE COMMENTS IN YOUR FLOWCHARTS AND PROGRAMS. MOREOVER, USE DESCRIPTIVE VARIABLE NAMES. REMEMBER, READABILITY MATTERS.
- SUBMIT IT BY THE DUE DATE AGREED UPON. NO FURTHER EXTENSION SHALL BE ENTERTAINED.
- YOU MUST INCLUDE THE NON-PLAGIARISM DECLARATION FORM ALONG WITH YOUR SUBMISSION.
- AUTOMATED PLAGIARISM CHECKING TOOLS WILL BE EMPLOYED. REFRAIN FROM THE MALPRACTICE OF PLAGIARISM.

Mandatory Non-plagiarism Declaration Form

(You must include this page with the report, filled and completed)

Plagiarism is defined as the presentation by an individual of another person's ideas or work (in any medium, published or unpublished) as though they were his/her own.

(Section 2.2 of the Student Handbook, 2017)

- I understand that plagiarism is an academic dishonesty and an offense that will be penalized, including failing in the course and even suspension from the college.
- I hereby declare that all the work in this assignment is all mine and I have not copied from anyone or anywhere to the best of our knowledge. I am well-aware of the seriousness of plagiarism and its severe consequences if caught in this offense, as stated in the college's regulation.

Roll No: _____

Date: _____

- I acknowledge that I have used the following resources/references for this assignment. Explicitly write ALL references (if any) in proper format.

Deliverable 1: Algorithm / Flowchart

1. Write an algorithm for **any three** of the following questions.
 - (a) Find the maximum of 4 user-supplied numbers.
 - (b) Display the sum of the numbers between 1 and N that are *divisible by 3* (where N is a user-supplied number).
 - (c) Reverse the digits of a number and display the reversed number. E.g. 1234 becomes 4321, 6932 becomes 2396
 - (d) Find the GCD and LCM of two numbers accepted from the user.
2. Design a flowchart for **any three** of the above questions using RAPTOR.

Deliverable 2: Python program

1. Write *menu-based* program for the questions in deliverable 1 i.e. (a)-(d). Your program must display 4 menu items the user shall choose from:


```
[1] Max of 4 numbers
[2] Add divisible-by-3 numbers within a range (1 to N).
[3] Reverse a number
[4] Compute GCD and LCM
```

 - Input other than 1-4 must be treated as invalid input. If a user selects a specific item (e.g. 1), perform the chosen operation and then let the user decide if s/he wants to proceed to another operation or exit.
 - Use of your own functions is encouraged although not mandatory for this question. Make it a habit to write comments throughout your programs.
2. Write a function-based program to calculate grades of the modules you are taking this semester. Your program must have the following 3 functions defined:
 - **Get_marks()**: is a function to get from the user both CA marks and final exam results of all the 5 modules you are taking.
 - **Calculate()**: is a function that uses the inputs from the previous function and performs various computations.

- i) Total mark for each module (out of 100).
 - ii) The average mark of all the modules.
 - iii) The lowest mark of all the modules.
 - iv) The status of your performance in each module according to RUB's specification (see section D1 of the Wheel at <http://www.rub.edu.bt/regulation/>).
- **Display()**: is a function that just prints the results of the previous **computation()** function.
3. Using functions, write *both iterative and recursive* programs for each of the following.

- (a) **DoubleFactorial(n)**: calculates the double factorial of n. Double factorial of n (denoted by $n!!$) is defined as:

$$\begin{array}{ll} n (n-2) (n-4) \dots 4*2 & \text{(for even } n) \\ n (n-2) (n-4) \dots *3*1 & \text{(for odd } n) \end{array}$$

$$\text{E.g. } 6!! = 6*4*2 \text{ and } 9! = 9*7* \dots *3*1$$

- (b) **Fibonacci(x)**: generates the first 'x' members of a Fibonacci sequence. In this sequence (starts from 0 and 1), each number is the sum of the the preceding two numbers. It looks like:

$$0, 1, 1, 2, 3, 5, 8, 13, \dots$$

- (c) **Combination (n, r)**: computes number of combination when 'r' items are selected from 'n' items without repetition. Recall from your highschool maths that $nCr = n! / r! (n-r)! .$