Punjab University College of Information Technology (PUCIT)

University of the Punjab

Programming Fundamentals – Fall 2021

(BSDS F21 M&A Sections)

Google Classroom
BSDSF21PF

Course code	CC-102
Credit hours	3
Prerequisite	Nil (but Enthusiasm, Consistency and Honesty)
Follow up	CMP-201 Object Oriented Programming

Course Instructor Muhammad Idrees
Email: *sir.idrees@gmail.com (general)*
ta.bsdsf21pf@gmail.com (submissions)
sir.bsdsf21pf @gmail.com (copy submissions)
\\printsrv\Faculty Data\Muhammad Idrees\PF Fall 21 (BSDSF21)
Office hours: Tuesday/Thursday (11:30am to 12:15pm)
Tuesday/Thursday (01:15pm to 02:00pm)
Any other time (working day) upon permission through email

Course Objectives

- To introduce what computer programs are, how an existing program behaves.
- To develop the skills building in Python/C++ programming.
- To translate their program logic (basic pseudo-code/flow-charts) into some programming language that computer can understand so that they can get real feel of their efforts.
- To introduce how to edit/update existing programs and how to build relatively large programs.
- To basic principles of attacking a problem, a bit of performance factor and some basic structured design principles.

Textbooks

- Tony Gaddis, *Starting out with Python*, 5th Ed., Addison-Wesley.
- John V. Guttag, Introduction to Computation and Programming Using Python (with Application to Understanding Data), 2nd Ed., The MIT Press

Reference Books/Websites

- https://www.programiz.com/python-programming
- https://www.youtube.com/playlist?list=PLi01XoE8jYohWFPpC17Z-wWhPOSuh8Er-

C/CPP Books/Websites for further references

- John R. Hubbard, *Schaum's Programming with C++*, 2nd Edition, McGRAW HILL.
- Tony Gaddis, *Starting out with C++: from control structures through objects*, 7th Ed., Addison-Wesley.
- D.S. Malik, *C++ Programming, From Problem Analysis to Program*, Design, 5th Ed.
- http://www.learncpp.com/.
- https://www.youtube.com/playlist?list=PLAE85DE8440AA6B83

Grading Instruments (Sessional decomposition may vary at end of course)

5 to 8 marks for	Quizzes, planned/announced or sudden
10 to 15 marks for	Programming assignments
2 to 3 marks for	Written assignments
Up-to 5 marks for	Term project / technical report
35 marks for	Midterm exam/pre-mid lecture notes
40 marks for	Final exam/post-mid lecture notes

Passing Criteria

- As per college rules, minimum requirement to pass this course is to get overall 50% marks.

Tentative Course Outline and Lecture Plan

Topics	No. of Lectures
Introduction of Course, Introduction to Python, CMD vs IDE, Output in brief, Hello World, (bool, int, float, and string, …) objects in Python	1
Constants, Variables, Identifiers, Operators, Functions, and Expressions Simple UDTs: enumerations and structures, Intro of advanced types	2
Input in brief, Assignment operation, Type Casting functions, further discussion of operators (arithmetic, relational, logical, concatenation, …)	2
Selection; if, if else, if elif … else, single line if else	1
Iterations; while and for, range type	2
Functions: define and calling functions, formal and actual parameter, call by value and call by reference, local and global variables, SCOPE and LIFETIME	2
Lists; Homogenous and Heterogeneous lists; Nested Lists Passing lists to functions, returning lists	1
Exception handling	1
Combining learned concepts to develop relatively big programs	4
MIDTERM EXAM	
Text files handling	2
Formatted input/output, I/O in detail/depth	1
Lambda functions, passing and returning functions, first class object, callback	1
Strings in details	1
List, Tuples, Sets, and Dictionaries, Mutable and Immutable types Shallow vs Deep copy, iterators and iterables, ***aliasing***	2
Arrays in Python, arrays vs Lists, ***ctype*** arrays, ***numpy*** arrays	2
Multidimensional arrays; Binary file handling	2
Modules, creating and using modules, python standard modules (libraries)	2
Misc topics: recursion, backtracking, graphs, use of stack/queue, sockets, threads, Object Oriented Programming, ~~Introduction to CPP~~	3
FINAL EXAM	

Important Notes

- Academic integrity is expected of all students. Plagiarism or cheating in any assessment will result in at least an **F** grade in the course, and possibly more severe penalties.
- You bear all the responsibility for protecting your assignments from plagiarism. If anyone else submits your assignment or uses your code in his/her assignment, you will be considered equally responsible.
- The instructor reserves the right to modify the grading scheme/marks division and course outline during the semester.
- All code written in quizzes, assignments, homework's, and exams must be in ***Python***. Code must be intelligently documented (commented). Undocumented code may not be given any credit.
- You **may** use Geany or PyCharm or IDLE with NotePad++ writing your code OR choose one of you own choice.
- There is no makeup for a missed sessional grading instruments like quizzes, assignments, and home works.