Department of Data Science

University of the Punjab

Object Oriented Programming – Spring 2023

(BSDS F22 M&A Sections)

**Course code** CC-201

**Credit hours** 3

**Prerequisite** CC-102 Programming Fundamentals

(and Enthusiasm, Consistency and Honesty too)

**Follow up** CS-203 Artificial Intelligence

**Course Instructor** Muhammad Idrees

Email: *sir.idrees@gmail.com (general)*

Google Classroom code: **fqp7two**

Office hours: Any time (working day) with permission through email.

**Course Objectives**

* To equip the learner with the philosophy and necessary skills to formulate solutions of real world problems using object-oriented paradigm.
* Justify the philosophy of object-oriented design and the concepts of encapsulation, abstraction, inheritance, and polymorphism.
* Strong concepts of object manipulation and dynamic memory allocation within classes

**Textbook**

* Tony Gaddis, *Starting out with Python:* 5th Ed., Addison-Wesley.
* Dusty Phillips, Python 3, Object Oriented Programming, 3rd Edition, Packt Publishing.

**Reference Books/Websites**

* Mark Lutz, Programming Python, *O’REILLY*.
* https://pynative.com/python/object-oriented-programming/
* https://www.youtube.com/ and https://www.google.com/ appropriate topic search.

## 

**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.

**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.
* There is no makeup for a missed sessional grading instruments like quizzes, assignments, and home works.

**~~Tentative~~ Course Outline Covered**

|  |  |
| --- | --- |
| Topics | No. of Lectures |
| Introduction of the course; Motivation: from structured programming to object oriented programming (OOP), everything (tangible, mathematical, or conceptual) is object; OOP Terminology and real world examples | 1 |
| Introduction to user defined types, class and pass keywords, variables of user defined types, creating and accessing data members using character · (dot) | 1 |
| 4 pillars of OOP, Abstraction (information hiding or separation of interface from implementation), Encapsulation (data and function members of class/object in one unit, the object), Inheritance (re-use and extension of existing code) and Polymorphism (one function with different implementations for several subclasses), Introduction of UML class diagram | 1 |
| Member functions; class level and instance level members (data/functions); static members; ~~nameless objects~~ | 2 |
| Information Hiding: private data members, getters and setters, properties; read only and write only properties/data members; constructors, \_\_init\_\_ (also \_\_new\_\_ ); destructor ( \_\_del\_\_ ) | 1 |
| Function overloading and overwriting/overriding | 2 |
| Magic functions (dunders) as members of classes | 1 |
| Aggregation and Composition; has a relationship | 2 |
| Inheritance and Polymorphism; is a relationship; super() | 2 |
| Multiple inheritance | 1 |
| Abstract base classes, abstract methods; interfaces; standardization | 1 |
| Exceptions and exception handling | 2 |
| Several case studies of ADTs including Vectors, Polynomials, Queues, Matrices, Bills, Shapes, Students/Employees, etc | 3 |
| Revision of File handing in PythonText and Binary files |  |
| Serialization/Deserialization (pickle, ~~json~~), ~~CSV files readers/writers~~ | 2 |
| Database vs files; simple QL; Connectivity to databases | 2 |
| ~~GUI programming; event driven programming~~ | ~~2~~ |
| Final Exam | |
| ~~Revision of Structured Programming; Pitfalls of Structured Programming; Argument categories (e.g., positional arguments); Revision of Mutable and immutable objects~~ | ~~4~~ |
| ~~Concurrency and threads; network programming~~ | ~~1~~ |
| ~~Missed/Misc. topics; Revision~~ | ~~1~~ |