



Object Oriented Programming (OOP)

Course Code:	CS-212	Semester:	3rd
Credit Hours:	3+1	Prerequisite Codes:	
Instructor:		Class:	
Office:		Telephone:	
Lecture Days:		E-mail:	
Class Room:		Consulting Hours:	
Lab Engineer:		Lab Engineer Email:	
Knowledge Group:		Updates on LMS:	

Course Description:

This course involves the learning of Object Orientated techniques including, Classes, Objects, Constructors, ,Destructors, Data Hiding, Inheritance, Operator Overloading, Polymorphism, Friend Functions, Abstract Classes, Management of Class Source & Object Files and Data Encapsulation

Course Objectives:

Course Learning Outcomes (CLOs):

Upon successful completion of the course, the student should be able to:	PLO	BT Level*
CLO-1. Solve real-life problems		
CLO-2. Designing programs using object-oriented design techniques.		
CLO-3. Develop object-oriented program analysis (OOA) skills		
CLO-4. Write a simple and complex python object-oriented application		
* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Mapping of CLOs to Program Learning Outcomes

PLOs/CLOs	CLO1	CLO2	CLO3



Mapping of CLOs to Assessment Modules and Weightages (In accordance with NUST statutes)

To be filled in at the end of the course.

Assessments/CLOs	CLO1	CLO2	CLO3
Quizzes: 10-15 %			
Assignments: 5-10 %			
Midterms: 30-40 %			
End Semester Exam: 40-50 %			
Total : 100 %			

Books:

Text Book:

- Reference Books:**
1. The Object-Oriented Thought Process Matt A. Weisfeld, 2000
 2. Python Object-Oriented Programming: Build Robust and Maintainable Object-oriented Python Applications and Libraries, 4th Edition by Steven F. Lott, 2021
 3. Gary Benson and Roderic Page, "Algorithms in Bioinformatics".
 4. Object-Oriented Python (Master OOP by Building Games and GUIs) by Irv Kalb

Sr. No	Main Topics to be covered	Week	Lecture wise Break up
1	Introduction		
2	Objects in Python <ul style="list-style-type: none"> • When Objects Are Alike • Expecting the Unexpected 		
3	When to Use Object-Oriented Programming <ul style="list-style-type: none"> • Procedural Python Examples 		
4	Modelling Physical Objects with Object-Oriented Programming		
5	Mental Models of Objects and the Meaning of "self"		
6	Managing Multiple Objects; Python Data Structures		
7-8	Conversion And Casting Abstract Base Classes and Operator Overloading		
8	MIDTERMS		
9	The Intersection of Object-Oriented and Functional Programming		
10	Graphical User Interfaces With Pygame <ul style="list-style-type: none"> • Object-Oriented Pygame • Pygame GUI Widgets 		
11	Strings, Serialization, and File Paths		
12	The Iterator Pattern Common Design Patterns		
13	Advanced Design Patterns		
14	Encapsulation, Polymorphism, And Inheritance <ul style="list-style-type: none"> • Managing Memory Used by Objects 		



15	Testing Object-Oriented Programs		
16	Python, Java, and C++: a Transition Guide		
17	END SEMESTER EXAMINATION		

Grading Policy:	
Quiz Policy:	The quizzes will be unannounced and normally last for ten minutes. The question framed is to test the concepts involved in last few lectures. Number of quizzes that will be used for evaluation is at the instructor's discretion. Grading for quizzes will be on a fixed scale of 0 to 10. A score of 10 indicates an exceptional attempt towards the answer and a score of 1 indicates your answer is entirely wrong but you made a reasonable effort towards the solution. Scores in between indicate very good (8-9), good (6-7), satisfactory (4-5), and poor (2-3) attempt. Failure to make a reasonable effort to answer a question scores a 0.
Assignment Policy:	In order to develop comprehensive understanding of the subject, assignments will be given. Late assignments will not be accepted / graded. All assignments will count towards the total (No 'best-of' policy). The students are advised to do the assignment themselves. Copying of assignments is highly discouraged and violations will be dealt with severely by referring any occurrences to the disciplinary committee. The questions in the assignment are meant to be challenging to give students confidence and extensive knowledge about the subject matter and enable them to prepare for the exams.
Plagiarism:	SINES maintains a zero tolerance policy towards plagiarism. While collaboration in this course is highly encouraged, you must ensure that you do not claim other people's work/ ideas as your own. Plagiarism occurs when the words, ideas, assertions, theories, figures, images, programming codes of others are presented as your own work. You must cite and acknowledge all sources of information in your assignments. Failing to comply with the SINES plagiarism policy will lead to strict penalties including zero marks in assignments and referral to the academic coordination office for disciplinary action.

Labs	Description
1	Setting up environment to edit, compile and run python programs
2	Input and Output in Java, understanding core elements of a python program
3	Classes and Objects, Access Modifiers
4	Variables, Data types, Writing expressions
5	Program Flow, Selection and Repetition Control Structures
6	Working with Arrays



7	Inheritance, Super-classes and Sub-classes
8	Polymorphism and Dynamic Method Binding
9	MID SEMESTER EXAMS
10	Creating a Graphical User Interface (GUI) using Python AWT and Swing
11	Event Handling in python
12	Pattern Matching and Regular Expressions in python, Applications in Bioinformatics
13	Third Party APIs and Integration for building Bioinformatics Applications
14	Creating a simple application pipeline using python
15	Student Project Demonstration
16	

Sr. No	Main Topics to be covered	Estimated Contact Hours
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		