COURSE OUTLINE
SCHOOL OF ENGINEERING AND APPLIED SCIENCES

Object-Oriented Programming (CS – 239)
BS Computer Science / BS Software Engineering

Spring 2020 Semester

Faculty:	School of Engineering and Applied Sciences
Credit Hours:	3
Semester	Spring 2020
Course Convenor:	Kamal Ashraf Gill
WhatsApp	+92-332-6201400
Email	kamal.ashraf@gift.edu.pk
LMS Link	
Consultation Hours:	TBD
Pre-requisite:	CS–123: Introduction to Programming
Core/Elective:	Core
Timing	Monday, 08:00-09:15 Friday, 08:00-09:15
This document was last updated:	June 23, 2020 V1.2

Course Description

This course teaches object-oriented programming to those who have learnt basic programming concepts and are ready to learn in-depth programming. It focuses on object-oriented programming using Java. The main concepts discussed are: Abstractions, Objects, Data Abstraction, Data Encapsulation, Polymorphism, and Inheritance. We teach the Java language constructs that are used to implement these concepts. For example, Dynamic Memory, Classes, Overloading Methods and Constructors, Overriding Methods, Class Associations, Abstract Methods, and Generics.

Course Goals

The goals of this course are that students:

- Become familiar with the importance of abstractions
- Become familiar with breaking down a problem into objects rather than procedures
- Learn object-oriented programming using Java
- Learn the concept of thinking in terms of objects while creating programs
- Have a basic understanding of UML and UML Class Diagrams

Learning Outcomes

On successful completion of this course, students should be able to:

- Learn to appreciate and use abstraction, and abstract data types.
- Explain the benefits of basic object-oriented design and understand when it is an appropriate methodology to use.
- Design object-oriented solutions for small systems involving multiple objects.
- Understand the use of dynamic memory in the creation of strings, arrays, and objects.
- Understanding object composition and class associations.
- Understand and appreciate generic programming for creating generic types.
- Understand the use of UML to draw and visualize Class Diagrams and Object Associations.

Reading Material

Textbook:

Tony Gaddis, *Starting out with Java: From Control Structures through Objects,* 6th Edition, Pearson 2016.

Reference Books:

Kathy Sierra & Bert Bates, Head First Java, 2nd Edition, O'Reilly 2005.

Herbert Schildt, Java A Begginer's Guide, 7th Edition, Oracle Press 2018.

Doug Lowe, Java ALL-IN-ONE, 5th Edition, Wiley 2017.

Term Project (If any)
In this course, the students will do a project using OOP Class relationships.

COURSE SCHEDULE & CONTENTS

Class Session	Topics To Cover	Teaching Material / Delivery Mode	Reading Material	Assessment Due
1-2	• Course Introduction • Revision of methods, arrays, and passing arrays as method arguments	Slides, Notes, Code Examples / Online session	Chapter, 5, 7 (Methods, Arrays)	Homework 1
3-6	• Linear Search Algorithm • The Binary Search Algorithm • The Selection Sort Algorithm • Two Dimensional Arrays • Matrices and Matrix Operations File I/O Handling	Slides, Notes, Code Examples / Online session	Chapter 7 (Arrays)	Quiz 1 Assignment 1
7-10	• Abstract Data types and Introduction to Classes • Objects and Object References • Instance Fields and Methods	Slides, Notes, Code Examples / Online session	Chapter 6 (Objects and Classes)	Homework 2 Quiz 2 Project Assessment - 1

	• Constructors • Overloading Constructors and Methods • Passing Objects as Arguments • Scope of Instance Fields Introduction to UML Class Diagram			
11-15	• Passing Objects as Method Arguments • Returning Objects from Methods • The **toString** Method • Writing an **equals** Method • **this** Reference • Methods that Copy Objects • Shallow vs. Deep Copy • Writing Copy Constructors • Static Class Members Packages and import Statement	Slides, Notes, Code Examples / Online session	Chapter 7, 8 (Arrays, Classes in Depth)	Quiz 3 Project Assessment - 2 Assignment 2
16	**MID-TERM EXAM**			
17-20	• Generalization and Specialization • "Is-A" Relationship • UML Notation for Inheritance • Superclass Constructors • Method Overriding	Slides, Notes, Code Examples / Online session	Chapter 10 (Inheritance)	Homework 3 Quiz 4

	• Class Hierarchies The class **Object**			
21	• Object Associations • Composition - "Has-A" Relationship • Association - "Uses" Relationship UML Notation for Object Associations	Slides, Notes, Code Examples / Online session	Notes (Object Associations)	Project Assessment – 3
22	• Abstract Classes and Abstract Methods • UML Notation for Abstract Classes • Interfaces – The Contract • UML Notation for Interfaces Implementing Multiple Interfaces	Slides, Notes, Code Examples / Online session	Chapter 10 (Abstract Classes)	Assignment 3 Quiz 5
23-24	• Polymorphism • The **instanceof** Operator Polymorphism and Interfaces	Slides, Notes, Code Examples / Online session	Chapter 10 (Abstract Classes)	Project Assessment – 4
25-27	• Object Associations • Composition - "Has-A" Relationship • Association - "Uses" Relationship UML Notation for Object Associations	Slides, Notes, Code Examples / Online session	Notes (Object Associations)	Project Assessment – 3

28	• Wrapper Classes • Wrapper Classes for Numeric Datatypes The **ArrayList** Class	Slides, Notes, Code Examples / Online session	Chapter 9 (Wrapper Classes)	Quiz 6
29-30	• Why Use Generics? • Generic Methods Generic Types	Slides, Notes, Code Examples / Online session	Chapter Notes (Generics)	Assignment 4
END TERM EXAM				

ASSESSMENT

Item	Assessment Task	Frequency	Weightage
1.	Quizzes/Online Activities	14	20%
2.	Homework and Assignments	8	20%
3	Project	1	20%
4.	Final Examination	1	40%

- ❖ **Please note that retake of *any* missed assessment <u>will not</u> be permitted**.

- **Course materials, announcements, all assessment items, and recorded lectures will be hosted on Google Classroom.**
- **NOTE: All quizzes, assignments, and project deliverables will have submission time limits. Late submissions will have penalties as marks deductions. Details of penalties will be given in each assessment item document.**
- **There will be a live QA session during each scheduled lecture. You are all expected to participate and ask questions. The frequency of the live sessions may be adjusted as per the student's feedback. Details of the live sessions will be announced on Google Classroom.**

Students must complete each component of the assessment to the satisfaction of the course instructor, and achieve an overall mark of **at least 40%** in order to pass the course. All components of the above assessment are compulsory, and must be completed in order to obtain a pass grade. Students are expected to perform satisfactorily in each item.

Course Policies and Rules

Participation in, and contribution to, class discussions will positively help you in understanding and applying the concepts learned during the lectures. Raise your hand if you have any question. Making any kind of disruption (e.g. side talks, continually come to class late, continually leaving class early, use of cell phones, etc.) in the class is strongly discouraged and may result in the removal of students from the lectures and/or labs.

COURSE COMMUNICATION

All communication in this course will be done via emails on your **official student email accounts only**, and you are expected to regularly monitor your email for any important announcements during the course.

ATTENDANCE

Attendance (minimum **80%**) is compulsory. Students having attendance below than this limit will not be permitted to attend the final exams.

Prepared By:	**Kamal Ashraf Gill**	**Signature**
Reviewed By:	**Dr. Muhammad Faheem**	**Signature**
Recommended By:	**Dr. Ziad Nayyer Dar**	**Signature**
Approved By:	**Dr. Qaiser S. Durrani**	**Signature**
Reviewed by QEC:	**Director QEC**	**Signature**
Final Approval By:	**Online Academic Council**	**Date**