Independent University, Bangladesh
Department of Computer Science & Engineering

Course Outline

Course Information:

Course Code: **CSE213 + L**	Course Title: **Object Oriented Programming – I**	Section: **1, 2, 3 & 4**

Date & Time Information

Year: **2021**	Semester: **Autumn**	Days:	Time:
		ST (Sec-1 Course)	**12:30 – 14:00**
		S (Sec-1 Lab)	**14:00 – 15:30**
		MW (Sec-2 Course)	**11:00 – 12:30**
		M (Sec-2 Lab)	**09:30 – 11:00**
		MW (Sec-3 Course)	**12:30 – 14:00**
		W (Sec-3 Lab)	**14:00 – 15:30**
		ST (Sec-3 Course)	**17:00 – 18:30**
		T (Sec-4 Lab)	**15:30 – 17:00**

Instructor Information

Instructor: **Subrata K. Dey**	Office: **6005-C**	Tutorial Hour: **On appointment**	Time: **On appointment**
Email: **subrata@iub.edu.bd**	Mob: **01552315651**		
Teaching Assistant: **N/A**	Office: **6005-C**	Office Hr: **N/A**	Time: **N/A**

Course Web Information

Course web:	**Google Classroom and Piazza**	Piazza Class Name:	**CSE 213 OOP-I (sec 1, 2, 3 & 4)** You will be enrolled by the instructor

Course Description:

This is an advanced programming course in computer science, with particular emphasis on advanced concepts and theories of object oriented programming. The main objective of this course is to enable students to design software using an object oriented approach. Understanding object oriented concepts and theories would inspire students in designing and developing desktop application software (both console and GUI based) using Java programming language.

Attendance Policy:

1. It is the student's responsibility to gather information about the assignments and covered topics during the missed lectures. Regular class attendance is required.
2. Tentative date and syllabus of quiz, midterm and final exam is already given here, however, announcements will be given ahead of time if there is any change. There is **NO** provision for make-up quizzes or examinations.

P.T.O

Administrative Policy:

1. The lecture notes, reading materials, codes, or other resources will be made available prior to the discussion on that material in class so that students may have a cursory look into the materials.
2. Class participation is vital for better understanding of programming concepts. Moreover, this is considered as an indicator of a good learner. Students are encouraged to participate in raising relevant issues and discussing solutions to those questions. Students may be awarded bonus marks for raising interesting issues object oriented programming and providing answers to such questions.
3. Students are invited to raise questions at any point during the session.
4. Students should take tutorials during the schedule mentioned above. If necessary, prior appointment is required.

Academic Dishonesty:

1. A student who cheats, plagiarizes, or furnishes false, misleading information in the course is subject to disciplinary action up to and including an F grade in the course and/or suspension/expulsion from the University.
2. Students must maintain the code of IUB.
3. The goal of homework is to give you practice in mastering the course material. Consequently, you are encouraged to collaborate on problem sets. In fact, students who form study groups generally do better on exams than do students who work alone. If you do work in a study group, however, you owe it to yourself and your group to be prepared for your study group meeting. Specifically, you should spend at least 30-45 minutes trying to solve each problem beforehand by yourself. If your group is unable to solve a problem, talk to other groups or ask your recitation instructor or teaching assistant assigned to your class.
4. You must write up each problem solution by yourself without assistance. It is a violation of this policy to submit a problem solution that you cannot orally explain to a member of the course staff.
5. **No collaboration whatsoever is permitted during examination.**
6. Plagiarism and other anti-intellectual behavior cannot be tolerated in any academic environment that prides itself on individual accomplishment. If you have any questions about the collaboration policy, or if you feel that you may have violated the policy, please talk to one of the faculty.

Non-Discrimination Policy:

The course and University policy prohibits discrimination on the basis of race, color, religion, sex, marital or parental status, national origin or ancestry, age, mental or physical disability, sexual orientation, military status. If you see either by course instructor or any other person related to course showing any form of discrimination, please inform the proctor’s office of the wrong doing.

Teaching Method:

Because this course is heavily application oriented every week there are two a 90-minute theory classes and one 90-minute tutorial/lab session conducted where the students will get to know how to apply the theories learned during the theoretical lecture hours with the aid of Java programming language.

P.T.O

Evaluation Method and Marks Distribution:

Students are evaluated based on their ability to apply knowledge acquired during the course.

Type of Assessment	**%**	**Remark**
Midterm Exam **[23-OCTOBER-2021]**	15%	Tentative date: **[23-OCTOBER-2021]**
Final Exam **[11-DECEMBER-2021]**	35%	Tentative date: **[11-DECEMBER-2021]**
Quizzes/Lab Tasks	10%	Dates will be announced in previous class
Project Milestones	35%	You need to meet deadlines of various milestones
Attendance	5%	Will be calculated from iras attendance
Total	**100**	

Course Project Details:

It is an INDIVIDUAL course project & each of you will be given a unique project-topic. In this project, you are required to analyze requirements, then design and implement a GUI-based Java-FXML application software using an Object-Oriented design approach to meet a real-world business operation. Your Course Project will be evaluated based on the following:

Milestones (MS)	**Remark**
MS-1: Customer Requirement Analysis	You need to do some research as guided by the instructor to understand and finalize various requirements of different user-types of the software.
MS-2: System Design	You need to design a UML-Class-diagram for Model classes + File system for your project. Should meet the requirements finalized in MS-1.
MS-3: Designing Graphical User Interface (FXML Scenes)	Adequate Scenes should be provided to fulfill user requirements finalized in MS-1. The Scene-controller classes should be able to communicate with the Model classes of MS-2 (i.e. sending data while calling methods of Model classes from methods of Controller classes and returning data back to Controller class methods)
MS-4: Implementation	You need to write all Model classes specified in MS-2, all controller-classes specified in MS-3, using the binary & text files finalized in MS-2.
Viva	You need to explain all the milestones and give demo of your project. The acquired knowledge applied in your project thus-far will also be tested by relevant QA. You may also be asked to write a sample code for a live modification during viva.
Total	**If you fail to establish your own contribution (originality and authenticity of the work), you will not get any mark in your project. You MUST refrain yourself to take external help and submit something which is not done by you.**

P.T.O

Grading Policy:

A	A-	B+	B	B-	C+	C	C-	D+	D	F
90-100	84-89	80-84	75-79	70-74	65-69	60-64	55-59	50-54	45-49	0-44

* Numbers are inclusive

Reference Text Book(s):

1. Ivor Horton, *Ivor Horton's Beginning Java,* [Latest edition available in market], Wrox.
2. Herbert Schildt, *Java The Complete Reference*, [Latest edition available in market]
3. Bruce Eckel, *Thinking in Java,* MindView inc.
4. http://www.cplusplus.com/
5. https://docs.oracle.com/javase/tutorial/

Audit:

IUB Students who are willing to audit the course are welcome and are advised to contact the instructor before to check availability of space.

P.T.O

Class Schedule, Topics and Readings:

Class No	Major Topic of Discussion	Learning Outcome	Remark
Week 01 01	Overview and Introduction to OOP	1. Teaching & Learning plan, Assessment 2. Overview of class and object 3. Abstraction, encapsulation, data hiding	Course Outline, Google Classroom, OOP Lecture 00
02	Introduction to OOP (contd.)	1. Function/Method chaining 2. Constructors and destructors 2. Operator overloading 3. Method/Function overloading	Date-wise Lecture-note/Source-code-bundle 01
03	Lab Work for Class 01 and 02	1. Hands on practice on Class 01 and 02	
Week 02 04	Introduction to OOP (contd.)	1. Intro to Java. Understanding JDK, JRE & JVM 2. Features of Java. C++ vs. Java 3. Introduction to Java Package	Date-wise Lecture-note/Source-code-bundle 02
05	Introduction to OOP (contd.)	1. Review of control structures in using Java 2. *static*, *this*, super, *final* keywords in Java 3. *instanceof* operator 4. Java inner class	Date-wise Lecture-note/Source-code-bundle 03
06	Lab Work for Class 04 and 05	1. Hands on practice on Class 04 and 05	
Week 03 07	Console I/O & Random Numbers in Java	1. Various ways of taking input from console and displaying output (Scanner and System class) 2. Random numbers in Java	Date-wise Lecture-note/Source-code-bundle 04
08	Introduction to Java Array	1. Understanding Array from Java's perspective 2. Declare and allocate memory for multi-dimensional arrays 3. Introduction to ArrayList and advanced for loop	Date-wise Lecture-note/Source-code-bundle 05
09	Lab Work for Class 07 and 08	1. Hands on practice on Class 07 and 08	
Week 04 10	Aggregation & Inheritance	1. Aggregation and its use 2. Inheritance & its use (why?) 3. Types of Inheritance 4. Method Overriding	Date-wise Lecture-note/Source-code-bundle 06
11	Inheritance (contd.)	1. Abstract class 2. Use of Constructor in Inheritance 3. Interface and its use in Inheritance	Date-wise Lecture-note/Source-code-bundle 07
12	Lab Work for Class 10 and 11	1. Hands on practice on Class 10 and 11	
Week 05 13	File I/O in Java	1. Exception handling & its advantage 2. Hierarchy of Java Exception classes 3. Types of Exception	Date-wise Lecture-note/Source-code-bundle 08
14	File I/O (contd.)	1. Checked & unchecked exception 2. try, catch, throw, throws, finally keywords	Date-wise Lecture-note/Source-code-bundle 09
15	Lab Work for Class 13 and 14	1. Hands on practice on Class 13 and 14	
Week 06 16	Exception Handling	1. File I/O. Difference between text and binary file 2. Reading, writing and appending to a text file	Date-wise Lecture-note/Source-code-bundle 10
17	Exception Handling (contd.)	1. Reading, writing and appending to a binary file 2. Hands on practice on Class 16 & 17	
18	**Midterm Exam**		

Week 07 19	Introduction to Multithreading in Java	1. Introduction of Thread and multitasking 2. Life cycle of a thread, creating thread	Date-wise Lecture-note/Source-code-bundle 11
20	Concurrency: Multithreading for Java console application	1. Extending Thread class vs. implementing Runnable interface and understanding their context of usage	Date-wise Lecture-note/Source-code-bundle 12
21	Lab Work for Class 19 and 20	1. Hands on practice on Class 19 and 20	
Week 08 22	Graphical User Interface using Java FXML	1. Introduction to Java FX 2. Discussion on MVC architecture 3. Understanding Stage, Scene & FXMLController 4. Discussion on FXML classes & SceneBuilder	Date-wise Lecture-note/Source-code-bundle 13
23	Java FXML (contd.)	1. Understanding FXML controls: Label, TextField, TextArea, PasswordField, Button 2. Understanding different FXML containers/layouts 3. Using above components in Java FXML application	Date-wise Lecture-note/Source-code-bundle 14
24	Lab Work for Class 22 and 23	1. Hands on practice on Class 22 and 23	
Week 09 25	Java FXML (contd.)	1. Understanding FXML controls: CheckBox, RadioButton, ComboBox, 2. Using above components in Java FXML application	Date-wise Lecture-note/Source-code-bundle 15
26	Java FXML (contd.)	1. Understanding Table in FXML Application 2. Populating table from other FXML controls 3. Spawning new Stage and Scenes at runtime 4. Sending data from one Scene to another Scene	Date-wise Lecture-note/Source-code-bundle 16
27	Lab Work for Class 25 and 26	1. Hands on practice on Class 25 and 26	
Week 10 28	Java FXML (contd.)	1. Loading data into FXML controls including table from file (both text file and binary file) 2. Processing table data based on the selection of rows of the table (reading cells/ editing, etc) 3. Save and open file using FileChooser 4. Displaying different types of dialog boxes	Date-wise Lecture-note/Source-code-bundle 17
29	Java FXML (contd.)	1. Understanding Menu, MenuBar & MenuItem 2. Using Menu, MenuBar & MenuItem in Java FXML application	Date-wise Lecture-note/Source-code-bundle 18
30	Lab Work for Class 28 and 29	1. Hands on practice on Class 28 and 29	
Week 11 31	Java FXML (contd.)	1. Understanding Charts in Java FXML 2. Generating different types of charts processed data from FXML controls	Date-wise Lecture-note/Source-code-bundle 21
32	Java FXML (contd.)	1. Generating PDF with processed data from FXML controls.	Date-wise Lecture-note/Source-code-bundle 22
33	Lab Work for Class 31 and 32	1. Hands on practice on Class 31 and 32	
Week 12 34	Review	1. Review on Final Exam 2. Review on course-project demo	
35	**Review for final exam**		
36	**Final Exam**		