CSE111: Programming Language II

A. Course General Information:

Course Code:	CSE111
Course Title:	Programming Language II
Credit Hours (Theory+Lab):	3 + 0
Contact Hours (Theory+Lab):	3 + 3
Category:	Program Core
Type:	Required, Engineering, Lecture + Laboratory
Prerequisites:	CSE110 Programming Language I

B. Course Catalog Description (Content):

This course would introduce data structures, formal specification, and syntax of Object Oriented Programming (OOP), elements of language theory, and mathematical preliminaries. Other topics that would be covered are formal languages, structured programming concepts, and a survey of features of existing high-level languages. Students would design and write applications using an appropriate language. The course includes a compulsory 3-hour laboratory work each week.

C. Course Objective:

The objectives of this course are to

a. contrast between Procedural Programming and Object-Oriented Programming(OOP) and demonstrate the way OOP divides complex problems into smaller parts encouraging modular development.
b. introduce the students to the fundamental concepts of Object-Oriented Programming Language like classes, objects, and their properties, and apply them to model real-world applications.
c. enlighten the students about the differences between different types of access modifiers and their importance in encapsulation.
d. explain the rudimentary principles of inheritance and polymorphism and how they can be applied to minimize the complexity of an object-oriented program through code reusability and make it scalable.

e. familiarize the students with the concept of abstract classes, abstract methods, and their application in setting standards for other classes inheriting them.

D. Course Outcomes (COs):

Upon successful completion of this course, students will be able to

Sl.	CO Description	Weightage (%)
CO1	**Understand** the fundamental entities of Object-Oriented Programming(OOP), including the concepts of class, object, constructor, instance variables, and methods, and how they are used to model real-world entities and their behaviors.	40%
CO2	**Infer** outputs from the logical sequence of a program consisting of a combination of OOP attributes and methods satisfying the fundamental concepts of procedural programming.	20%
CO3	**Apply** access modifiers such as private, public, and protected to control the visibility of instance properties and **use** class-level variables and methods, to demonstrate their importance in ensuring data encapsulation, information hiding, and memory management.	15%
CO4	**Acknowledge** the basics of inheritance in Object-Oriented Programming, including the notions of variable and method overriding, and inheritance hierarchy, and how they can be used to create more complex and scalable programs ensuring proper code reusability.	25%

E. Mapping of CO-PO-Taxonomy Domain & Level- Delivery-Assessment Tool:

Sl.	CO Description	POs	Bloom's taxonomy domain/level	Delivery methods and activities	Assessment tools
CO1	**Understand** the fundamental entities of Object-Oriented Programming(OOP), including the concepts of class, object, constructor, instance variables, and methods, and how they are used to model real-world entities and their behaviors.	a	Cognitive/ Understand	Lectures, Notes, Lab class	Midterm Exam, Final Exam, Lab work
CO2	**Infer** outputs from the logical sequence of a program consisting of a combination of OOP attributes and methods satisfying the fundamental concepts of procedural programming.	a	Cognitive/ Understand	Lectures, Lab class	Midterm Exam, Final Exam, Lab work
CO3	**Apply** the usage of proper access modifiers such as private, public, and protected to control the visibility of instance properties and use class-level variables and methods, to demonstrate their importance in ensuring data encapsulation, information hiding, and memory management.	b	Cognitive/Apply	Lectures, Notes, Lab class	Final Exam, Lab work

CO4	**Analyze** the basics of inheritance in Object-Oriented Programming, including the notions of variable and method overriding, and inheritance hierarchy, and how they can be used to create more complex and scalable programs ensuring proper code reusability.	b	Cognitive/Apply	Lectures, notes, Lab class	Final Exam, Lab work

F. Course Materials:

i. Text and Reference Books:

Sl.	Title	Author(s)	Publication Year	Edition	Publisher	ISBN
1	Java: The Complete Reference	Herbert Schildt	2021	12th ed.	McGraw Hill	ISBN-13: 978-1260463415
2	Head First Java	Kathy Sierra, Bert Bates, Trisha Gee	2022	3rd ed.	O'Reilly Media	ISBN-13: 978-1491910771
3	Introduction to Java Programming and Data Structures, Comprehensive Version	Y. Daniel Liang	2019	12th ed.	Pearson	ISBN-13: 978-0136520238

ii. Other materials (if any)

a. Lecture notes and presentation slides
b. Lab sheets
c. Lab usage manual
d. IDE: Dr. Java

G. Lesson Plan:

No	Topic	Week/Lecture#	Related CO (if any)
1	Java Basics and Fundamental of Collection (Single and Multi-dimensional array)	Week 1,2	
2	Introduction to OOP, Class, object	Week 3	CO1
3	Instance Variables & Instance Method	Week 4	CO1

4	Overloading & Tracing object-oriented programs	Week 5	CO1,CO2
5	Data encapsulation and access modifiers.	Week 6	CO1,CO3
Midterm Exam			
6	Static variables and methods	Week 7	CO3
7	Basics of inheritance	Week 8	CO4
8	Method overriding and Tracing object-oriented programs	Week 9	CO4, CO2
9	Polymorphism	Week 10	
10	Abstractract class and method	Week 11	
11	Interface	Week 12	
Lab	Practice using Modern Tools	Lab classes	CO1, CO2, CO3, CO4
Final Exam			

H. Assessment Tools:

Assessment Tools	**Course Outcome**	**Weightage (%)**
Quizzes and Assignments	-	20
Midterm Exam	CO1, CO2	25
Final Exam	CO1, CO2, CO3, CO4	30
Lab including Lab Quizzes	CO1, CO2, CO3, CO4	25

I. CO Attainment Policy:

As per the Department of CSE Course Outcome Attainment Policy

J. Grading policy:

As per Brac University's grading policy

k. Course Coordinator:

Md. Tawhid Anwar & Md. Sabbir Ahmed