# ABET -2000
# COURSE BINDER

# CSC-122

# Object Oriented
# Programming

# PREPARED BY

## Dr. Sher Muhammad Doudpota

## Dr. Ghulam Mujtaba Sheikh

## Dr. Fahim Akhtar Rajput

## Spring 2024

## Course Syllabus (ABET -2000 Format):

### CSC-122 – Object Oriented Programming
### Spring 2024

| | |
|---|---|
| **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. In just a few years, java grew from a concept into one of the world's dominant computer language.<br><br>The main topics discussed are: Moving from C to Java, Class design and objects, Data Abstraction, Encapsulation, Polymorphism and Inheritance, Interfaces, Exceptional Handling, Multi-Threading, GUI Programming and Java Database Connectivity. |
| **Prerequisite** | Introduction to Programming (Using C Language) |
| **Textbook** | 1. Java 2 Complete Reference by Herbert Schildt, Latest edition<br>2. Beginning Java 2 by Ivor Horton, Latest edition<br>3. Absolute Java by Walter Savitch and Kenrick Mock, Fifth Edition |
| **Course Learning Objectives (C.L.O.)** | 1. Student should know about the object-oriented programming<br>2. Become familiar with breaking down a problem into objects rather than procedures<br>3. Inheritance and polymorphism<br>4. Generic programming and interface<br>5. Object oriented software development<br>6. Learn what is available off the shelf to facilitate Java development (tools, libraries)<br>7. Graphic user interface, event handling, and animation<br>8. Socket programming<br>9. Student should have good command over the syntax of JAVA<br>10. Student should submit a project designed by him/her-self by properly implementing all the concepts learned during semester |

| Course Relationship to Program Outcomes | Program Outcomes for Computer Science | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Highest attainable level of Learning | a | b | c | d | e | f | g | h | i | j | k |
| | | 1 | 2,6 | 1 | 1 | 1,10 | 1 | 1 | 1 | 1 | 1,10 | 1,10 |

| Class/ Lab Schedule | Theory – 3 classes in a week<br>Laboratory – 3 classes in a week<br>*(Class duration is 50 Minutes)* |
|---|---|
| Instructional Methods | Lectures – Tutorials – Home works – Mid. Exams – Final Exam. |
| Instructor | Dr. Sher M. Doudpota, Dr. Fahim Akhtar Rajput, Dr . Ghulam Mujtaba Sheikh |

# CSC-122
## Object Oriented Programming

| Course Learning Objectives (C.L.O.) | Program Outcomes | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | a | b | c | d | e | f | g | h | i | j | k |
| 1. Student should know about the object oriented programming | | | | | | | | | | M | M |
| 2. Become familiar with breaking down a problem into objects rather than procedures | | | | L | | | | | | | |
| 3. Inheritance and polymorphism | | | | | | | | | | | L |
| 4. Generic programming and interface | | | | | | | | | | | L |
| 5. Object oriented software development | | | | | | | | | | | L |
| 6. Learn what is available off the shelf to facilitate Java development (tools, libraries) | | M | | | | | | | | | |
| 7. Graphic user interface, event handling, and animation | | | | | | | | | | | L |
| 8. Java Database Connectivity | | | | | | | | | | | L |
| 9. Student should have good command over the syntax of JAVA | | | | | | | | | | | L |
| 10. Student should submit a project designed by him/her-self by properly implementing all the concepts learned during semester | | | | | L | | | | | L | H |

**H=3= High = Synthesis & Evaluation levels**
**M=2= Medium = Application & Analysis Levels**
**L=1= Low = knowledge & Comprehension Levels**

### Program Outcomes
(a) Knowledge of core computing, mathematics, and science fitting to the discipline.
(b) Ability to apply acquired knowledge in developing solutions using state-of-the-art methods, techniques, skills, and tools.
(c) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
(d) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
(e) An ability to function effectively on teams to accomplish a common goal
(f) An understanding of professional, ethical, legal, security and social issues and responsibilities
(g) An ability to communicate effectively with a range of audiences
(h) An ability to analyze the local and global impact of computing on individuals, organizations, and society
(i) Recognition of the need for and an ability to engage in continuing professional development
(j) An ability to use current techniques, skills, and tools necessary for computing practice.

# Course Calendar

| Week. No | Topic | Topic Detail |
|---|---|---|
| 1 | Moving C to Java | Procedural vs. OOP<br>Why OOP?<br>History of Programming Languages<br>Why JAVA?<br>History of JAVA<br>Buzzwords<br>JDK, JRE and JVM |
| 2 | Introduction to OOP | Programming Paradigm<br>Abstraction<br>Three OOP Principles<br>Encapsulation Inheritance &<br>Polymorphism Together |
| 3 | Introduction to Java | First Java application<br>A second Short Example<br>Two Control Statements (if and for)<br>Using Block Code<br>Lexical Issues<br>Programming Style<br>Java Documentation |
| 4 | Fundamental Elements of Language | Fundamental Elements of Java (Datatypes, Variables<br>and Arrays)<br>Primitive Types<br>Literals<br>String Literal<br>Escape Sequence<br>Scope and lifetime of Variables<br>Type Conversion and Casting<br>Automatic Type Promotion in Expressions<br>Arrays (One Dimension and Multi Dimension)<br>Uneven Multidimensional Arrays |
| 5. | Operators | Operators (Arithmetic, Bitwise, Relational and Logical)<br>Ternary operator<br>Control Statements (Selection, iteration, jump)<br>Introduction to classes |
| 7. | Classes & Methods | Constructor<br>new Operator<br>Parameterized Constructor<br>this keyword<br>Instance Variable Hiding<br>Garbage Collection<br>finalize () Method |

| | | |
|---|---|---|
| 8. | **Encapsulation:**<br>Closer look at Classes, Methods, Access Controllers & References | Methods<br>Overloading Methods<br>Overloading Methods – Automatic Conversion<br>Overloading Constructors<br>Using Objects as Parameters<br>Object to its Constructor as Parameter<br>A Closer Look at Argument Passing<br>Arguments in JAVA<br>Call by value<br>Call by Reference<br>Returning Objects<br>Recursion<br>Access Control (Specifiers: public, private<br>protected)<br>Understanding static and final<br>Nested Classes<br>Inner Class within any block scope |
| 9. | Inheritance | Inheritance<br>Extends<br>Supper & Subclass<br>Advantages of Inheritance<br>Extension in existing class<br>Reference a Subclass Object<br>Using *super*<br>Multilevel Hierarchy<br>Constructors Call<br>Method Overriding |
| 10. | Polymorphism | Method Overloading<br>Using Abstract Classes<br>3 usage of final<br>Packages and Interfaces<br>Packages<br>Compilation & Executing |
| 11 | Packages & Interfaces | Importing Packages<br>Creating & Importing packages<br>Interfaces<br>Defining an interface<br>Implementing Interfaces<br>Accessing through references<br>Polymorphic Methods<br>Partial Implementation of an interface<br>Interfaces Can Be Extended |
| 12. | Exception Handling and File IO | Exception Handling<br>Uncaught exceptions<br>Try, catch<br>Throw, throws and final<br>File I/O<br>Streams<br>Simple file I/O<br>Create a file<br>Add the record to the file<br>Read from existing file |

| | | |
|---|---|---|
| 14. | Graphical User Interface | Introduction to GUI<br>Awt & Swing<br>Steps for GUI Creation<br>First Java GUI Application<br>JFrame<br>Swing Components<br>2 different ways (Composition and Inheritance)<br>Layout Managers (FlowLayout, GridLayout, BorderLayout) |
| 15. | Java Database Connectivity | Introduction Database<br>Database Basics<br>SQL Basics<br>Four Kind of JDBC Drivers<br>JDBC-ODBC Bridge<br>Preparing Statement<br>Getting ResultSet<br>Updates in Database |

# CSC-122
## Object Oriented Programming

| Course Learning Objectives (C.L.O.) | Program Outcomes | | | | | | | | | | | Assessment Tools | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ab | c | d | | e | f | g | h | i | jk | | T1 | T2 | T3 | T4 | T5 | T6 |
| 1. Student should know about the object oriented programming | | | | | | | | | | | | | | | | | |
| 2. Become familiar with breaking down a problem into objects rather than procedures | | | | | | | | | | | | | | | | | |
| 3. Inheritance and polymorphism | | | | | | | | | | | | | | | | | |
| 4. Generic programming and interface | | | | | | | | | | | | | | | | | |
| 5. Object oriented software development | | | | | | | | | | | | | | | | | |
| 6. Learn what is available off the shelf to facilitate Java development (tools, libraries) | | | | | | | | | | | | | | | | | |
| 7. Graphic user interface, event handling, and animation | | | | | | | | | | | | | | | | | |
| 8. Java Database Connectivity | | | | | | | | | | | | | | | | | |
| 9. Student should have good command over the syntax of JAVA | | | | | | | | | | | | | | | | | |
| 10. Student should submit a project designed by him/her-self by properly implementing all the concepts learned during semester | | | | | | | | | | | | | | | | | |