Bangladesh Army University of Science and Technology

Department of Computer Science and Engineering

Course Outline: Summer Semester, 2020

Program: B.Sc. Engg. in Computer Science and Engineering

Level -2 Term-II **Total Credit: 3.00** **Credit Hour(s): 03**

Course No: CSE 2203 **Course Title: Object Oriented Programming II (Java)**

Course Objectives:

Students will learn

- ✓ The Java Programming Language: its syntax, idioms, patterns and styles.
- ✓ How to write, compile and execute Java programs using all major development tolls.
- ✓ To become comfortable with object oriented programming: Object and Class.
- ✓ How to deal with event driven Graphical User Interface (GUI) programming.
- ✓ To retrieve data from a relational database with Java Database Connectivity (JDBC)
- ✓ How to write Java programs that solve practical, real world, business-oriented problems.

Course Duration: 42 Hours (3 Hours per week).

Materials needed:

A Personal computer with JDK installed which includes a JRE along with Tools and APIs. Among with user interface toolkits (JavaFx, Swing, AWT etc.), libraries (integration libraries-Scripting, RMI, JDBC etc., language and util base libraries- math, collections etc. and other base libraries-security, networking etc.) JRE include a JVM which is java compiler that convert the source code into machine readable bytecode. You can also use a fully featured IDE preferably NetBeans.

Basic Text:

- Java - The Complete Reference: Ninth Edition - By Herbert Schildt.

Reference Books:

- Java How To Program (late objects) by Paul Deitel, Harvey Deitel
- Head First Java, 2nd Edition by Kathy Sierra, Bert Bates

Reference Website:

- www.javatpoint.com,https://www.tutorialspoint.com,
- https://docs.oracle.com/javase/tutorial/java/
- YouTube channel : PyAcademy2020

Marks Distribution:

Theory Course		Sessional Course	
Class Participation / Observation	5%	Class Attendance	10%
Class Attendance	5%	Class Performance	10%
HW/ Assignment/ Quizzes/**Class tests**	20%	Report	10%
Final Examination (3 hours)	70%	Quiz	20%
Total	**100%**	Viva	20%
		Lab Test	30%
		Total	**100%**

It is not enough just to write code that works. It is as important--perhaps more important to write code well; not merely code that works, but code that is legible, maintainable, reusable, fast, and efficient.

Lesson Plan

Week	Course contents	Remarks
01	**Introduction & Overview** • **History of JAVA** - The History and evolution of Java • **JAVA Development Environment** - JDK, JRE, JVM, API & Bytecode • **Features of Java:** Java Strength and Weakness • **The Three OOP Principal: Advances of JAVA** - Encapsulation, Polymorphism and Inheritance - A Sample Java Program – Compiling and Execution	
02	**Java Basics : POPs Concepts** • **Lexical issues** - White space, Comments, Separators, Identifier, and Keywords • **Data types, Variables and Literals** - The primitive types - integer, floating point, character & Boolean - Variables - declaring, initializing and lifetime of variables - Literals - - integer, floating point, character , boolean & string - Type conversion and casting, Automatic type promotion	
03	**Java Basics : POPs Concepts** • **Array** - One-dimensional, Multi-dimensional - alternative declaration - Array of uneven second dimension **Java Strings and Pointer**	
04	**Java Basics: POPs Concepts** • **Operators** - Arithmetic operators, Unary operator, Bitwise (logical and shift) operators, Relational operators, Boolean logical operators, Assignment operators, The Ternary(? :) operator, Precedence • **Control Statements** - Java's selection Statement – if else , switch - Java's Iteration Statement – for, while and do-while loop, For-each Java's Jump Statement – break, continue, return	CT 1
05	**Java OOPs Concepts – Beginner** • **Introduction to Classes** - Class Fundamentals – General form of a class - Declaring Object – the new keyword - Introducing methods – add method(s) to class, returning value from method(s) and parameterized method - Constructors – parameterized constructor - The **this** keyword – instance variable hiding - Garbage Collection and finalize () Method	
06	**Java OOPs Concepts – Beginner** • **A closer Look at Objects, Methods and Classes** - Overloading Constructor and Methods - Passing, Returning and assigning Object - Introducing Access Control - Understanding *static* and *final* - Varargs*:* Overloading Vararg methods, - Vararg and Ambiguity. - Command line argument	CT 2

It is not enough just to write code that works. It is as important--perhaps more important to write code well; not merely code that works, but code that is legible, maintainable, reusable, fast, and efficient.

07	• **String handling (The java library)** - String Constructor - String Constant pool - String classes and its methods - StringBuffer classes and its methods - and StringBuilder classes and its methods - String Vs. String Builder Vs. String Buffer	
Mid Term Break (*1 week*)		
08	**Java OOPs Concepts – Beginner** • **Inheritance** - Inheritance Basics – member access and inheritance - Using super – call superclass constructor, member hiding - Creating multi-level hierarchy - Execution sequences of Constructor - Method Overriding- Dynamic method dispatch - Final with inheritance – prevent overriding and inheritance - Using Abstract Class The Object class overview	
09	**Java OOPs Concepts – Intermediate** • ***Package* and Interfaces** - Defining Package - Importing Packages - Access Protection in packages - Packages and class path • **Package and *Interfaces*** - Defining and implementing interface - Interface variables - Default interface methods - Static Interface methods - Extending Interface	
10	**Java OOPs Concepts – Intermediate** • **Basic Error Handling: Exception Handling** - Exception-Handling Fundamentals - Exception Types - Java Exception classes Hierarchy - Checked and Unchecked exception - Uncaught exception - Using try and catch - Multiple catch, Nested try - Throw, Throws & finally - Java built-in Exception - Creating your own exception	
11	**Java OOPs Concepts – Advanced** • Multithreading - Lifecycle of Thread - The main thread , User Thread & Daemon Thread - Creating A thread : Two way - Thread Priorities, Synchronization - Inter thread communication –Dead lock - Suspend, Resuming and Stopping Threads	CT 3
12	**Java Generic, Lambda and GUI Concepts – Advanced** • Generics - General form of generics - Generics methods and - Generics interface	

It is not enough just to write code that works. It is as important--perhaps more important to write code well; not merely code that works, but code that is legible, maintainable, reusable, fast, and efficient.

	• Lamda Expression - What is Lamda Expression, - Block Lamda expression - Generic Functional Interface - Lambda Expressions and Arguments	
13	**Java Generic, Lambda and GUI Concepts – Advanced** • Java Collection Basics • Collection Interface • Java Collection Classes - ArrayList, Linked List - Priority Queue, ArrayDequeue - HashSet, TreeSet	
14	**Java Generic, Lambda and GUI Concepts - Advanced (if possible)** • Java Networking Basics - Networking concept - Socket programming • Basic concept of Java IO • Introducing GUI Programming - Java AWT, Swing, JavaFx - GUI Project – Calculator • Basic concept of Java JDBC **Java POP & OOPs Concepts** • Review Classes	

*The contents mentioned above are not definite rather students may experience continual change, provided such alterations are reasonable.

Course Policies:

Makeup Class or Class Test: In case of unavoidable circumstances should a course meeting, lecture or class test, needs to be cancelled; students will be offered a makeup class or class test rescheduled by course teacher.

Referred/ Improvement/ Backlog Exam: Students who get F grade in the final examination will get a chance to appear in a referred examination. The maximum grade of referred examination is B grade. Students have a chance to improve his cgpa by appearing in an improvement examination if he scores below B grade in the final examination. The maximum grade of improvement examination is B+ grade. Students who get F grade in the referred examination will get a chance to appear in a backlog examination in the next semester or later depending on the particular course is offered by. The maximum grade of backlog examination is C grade. Referred/ Improvement/ Backlog exam will be held 1/2 week after final exam. Please contact with the controller section and departmental head for overall detail updated exam policy.

Academic Policy: Any academic misconduct will result in an “F” grade for the course. There will be no exception.

Overall policy: Please consult the BAUST student code of conduct and guidelines offered by the registrar’s office.

Course Teacher

Md. Mamun Hossain
Assistant Professor,
Dept. of CSE, BAUST.
Email: mamunsust12@gmail.com

It is not enough just to write code that works. It is as important--perhaps more important to write code well; not merely code that works, but code that is legible, maintainable, reusable, fast, and efficient.