

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTHAPURAMU COLLEGE  
OF ENGINEERING (AUTONOMOUS):: PULIVENDULA**

<b>Course Code</b>	:				
<b>Course Title</b>	:	<b>Object Oriented Programming using Java</b>			
<b>Course Structure</b>	:	Lectures	Tutorials	Practicals	Credits
		3	1	0	3
<b>Course Coordinator</b>	:	Mrs. L.Bhavya			
<b>Team of Instructors</b>	:	Mr. G. Murali			

### I. Course Overview

Java programming language was originally developed by Sun Microsystems which was initiated by James Gosling and released in 1995 as core component of Sun Microsystems' Java platform (Java 1.0 [J2SE]). The latest release of the Java Standard Edition is Java SE 8.

### II. Prerequisite(s):

Level	Credits	Periods / Week	Prerequisites
UG	3	4	Basics of C, Object oriented concepts of C++

### III. Assessment:

<b>FORMATIVE ASSESMENT</b>	
Mid Semester Test I for 20 Marks in first 2 units is conducted at the end of 9 <sup>th</sup> week.	20 Marks
Mid Semester Test II for 20 Marks in last three units is	

conducted at the end of the course work.	
Average of two tests is taken as final	
Mid semester Test Multiple Choice Test in first two and half Units is conducted for 10 Marks	10 Marks
Mid semester Test Multiple Choice Test in second two and half Units is conducted for 10 Marks	
Average of two tests is taken as final	
Total ( Formative)	30 Marks
<b>SUMMATIVE ASSESMENT</b>	
End Semester Examination in all units is conducted for 70 Marks	70 marks
<b>Grand Total</b>	100 Marks

#### IV. Course objectives:

1. Study the syntax, semantics and features of Java Programming Language
2. Study the Object Oriented Programming Concepts of Java Programming language
3. Learn the method of creating Multi-threaded programs and handle exceptions
4. Learn Java features to create GUI applications & perform event handling

#### V. Course Outcomes:

1. Solve problems using object oriented approach and implement them
2. Ability to write Efficient programs that handle exceptions
3. Create user friendly interface

#### VI. Program outcomes:

##### Program Outcomes

- a An ability to apply knowledge of computing, mathematical foundations, algorithmic principles, and computer science and engineering theory in the modeling and design of computer-based systems to real-world problems (fundamental engineering analysis skills)
- b An ability to design and conduct experiments, as well as to analyze and interpret data (information retrieval skills)
- c An ability to design , implement, and evaluate a computer-based system, process, component, or program to meet desired needs, within realistic constraints such as economic, environmental, social, political, health and safety, manufacturability, and

sustainability (Creative Skills)

- d An ability to function effectively on multi-disciplinary teams (team work)
- e An ability to analyze a problem, identify, formulate and use the appropriate computing and engineering requirements for obtaining its solution (engineering problem solving skills)
- f An understanding of professional, ethical, legal, security and social issues and responsibilities (professional integrity)
- g An ability to communicate effectively both in writing and orally (speaking / writing skills)
- h The broad education necessary to analyze the local and global impact of computing and engineering solutions on individuals, organizations, and society (engineering impact assessment skills)
- i Recognition of the need for, and an ability to engage in continuing professional development and life-long learning (continuing education awareness)
- j A Knowledge of contemporary issues (social awareness)
- k An ability to use current techniques, skills, and tools necessary for computing and engineering practice (practical engineering analysis skills)
- l An ability to apply design and development principles in the construction of software and hardware systems of varying complexity (software hardware interface)
- m An ability to recognize the importance of professional development by pursuing postgraduate studies or face competitive examinations that offer challenging and rewarding careers in computing (successful career and immediate employment).

## VII. Syllabus:

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTHAPURAMU

COLLEGE OF ENGINEERING (AUTONOMOUS):: PULIVENDULA

B.Tech. (C.S.E)

L T P C

3 1 0 3

### Object Oriented Programming using Java

#### UNIT I

**The History and Evolution of Java:** Java's Lineage, The Creation of Java, How Java Changed the Internet, Java's Magic: The Bytecode, Servlets: Java on the Server Side, The Java Buzzwords, The Evolution of Java, Java SE 8, A Culture of Innovation.

**An Overview of Java:** Object-Oriented Programming, A First Simple Program, A Second Short Program, Two Control Statements, Using Blocks of Code, Lexical Issues, The Java Class Libraries.

#### UNIT II

**Data Types, Variables, and Arrays:** Java Is a Strongly Typed Language, The Primitive Types, Integers, Floating-Point Types, Characters, Booleans, A Closer Look at Literals, Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays, A Few Words About Strings, A Note to C/C++ Programmers About Pointers.

**Operators:** Arithmetic Operators, The Bitwise Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses.

**Control Statements:** Java's Selection Statements, Iteration Statements, Jump Statements.

#### UNIT III

**Introducing Classes:** Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, The finalize( ) Method, A Stack Class.

**A Closer Look at Methods and Classes:** Overloading Methods, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Recursion, Introducing Access Control, Understanding static, Introducing final, Arrays Revisited, Introducing Nested and Inner Classes, Exploring the String Class, Using Command-Line Arguments, Varargs: Variable-Length Arguments.

## UNIT IV

**Inheritance:** Inheritance Basics, Using super, Creating a Multilevel Hierarchy, When Constructors Are Executed, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, The Object Class.

**Packages and interfaces:** Packages, Access Protection, Importing Packages, Interfaces, Default Interface Methods, Use static Methods in an Interface, Final Thoughts on Packages and Interfaces.

**Exception Handling:** Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions, Three Recently Added Exception Features, Using Exceptions.

## UNIT V

**Multithreaded Programming:** The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive( ) and join( ), Thread Priorities, Synchronization, Interthread Communication, Suspending, Resuming, and Stopping Threads. Obtaining A Thread's State, Using Multithreading.

**Enumerations, Autoboxing, and Annotations (Metadata):** Enumerations, Type Wrappers, Autoboxing, Annotations (Metadata), Type Annotations, Repeating Annotations.

## VIII. List of Text Books / References / Websites / Journals / Others

### Text Books:

1. Java The Complete Reference 9th edition, Herbert Schildt, Mc Graw Hill Education, 2014.
2. Programming with Java, E. Balagurusamy.

### Reference Books:

1. J2SE Core Java, A.R.Kishore Kumar.

### Others:

1. Software Engineering Foundation, Trainee Guide

## IX. Course Plan:

The course plan is meant as a guideline. There may probably be changes.

Lecture Date	Course Learning Outcomes	Topics to be covered	Reference
Unit – I			
2/7/19	The History of Java	Java's Lineage, The creation of Java	T1:1:1,2 T2: 2:1
4/7/19, 5/7/19	Java with Internet and Servlets	How Java changed the Internet, Java's Magic: The Bytecode, Servlets: Java on the Server Side	T1:1:3-5 T2:2:4,5 R1: 2:7
9/7/19	Characteristics & Evolution of Java	The Evolution of Java, Java SE 8, A Culture of Innovation	T1:1:6,7,8 T2: 2:7,8,9
11/7/19	OOP's	Object Oriented Programming	T1:2:1 T2:1 R1:7:1,2 O1:1
12/7/19, 16/7/19	Sample program, Statements and Libraries	A First Simple Program, A Second Short Program, Two Control Statements, Using Blocks of Code, Lexical Issues, The Java Class Libraries	T1:2:2-7 R1:2:2-6
Practical sessions			
18/7/19, 19/7/19	Programming Techniques / Program thinking	Brief Discussion about how to analysis program	O2
Unit – II			
23/7/19, 25/7/19, 26/7/19	Data Types, Variables	Java Is a Strongly Typed Language, The Primitive Types, Integers, Floating-Point Types, Characters, Booleans, A Closer Look at Literals, Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions	T1:3 T2:4:4-7 R1:3:2
Practical sessions			
30/7/19, 1/8/19	Operators	Arithmetic Operators, The Bitwise Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses.	T1:4 T2:5:1-9 R1:3:3
Practical sessions			

2/8/19, 6/8/19	Control Statements	Java's Selection Statements, Iteration Statements, Jump Statements.  Practical sessions	T1:5 R1:4
8/8/19, 9/8/19	Arrays and Strings	Arrays, A Few Words About Strings, A Note to C/C++ Programmers About Pointers.  Practical sessions	T1:3:11,12,13 T2: 9:1-5 R1:6
Unit – III			
13/8/19, 16/8/19	Classes, Objects and Methods	Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods  Practical sessions	T1:6:1-4 T2:8:1-6 O1:2
27/8/19	Constructors, This and Finalize	Constructors, The this Keyword, The finalize ( ) Method, A Stack Class.  Practical sessions	T1:6:5-8 R1:7:4-7
29/8/19, 30/8/19	Method overloading and Argument Passing	Overloading Methods, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects  Practical sessions	T1:7: 1-4
3/9/19	Recursion, static and final	Recursion, Introducing Access Control, Understanding static, Introducing final.  Practical sessions	T1:7:5-8
5/9/19, 6/9/19	Classes	Arrays Revisited, Introducing Nested and Inner Classes, Exploring the String Class, Using Command-Line Arguments, Varargs: Variable-Length Arguments.  Practical sessions	T1:7:9-13 R1:8:4
Unit – IV			
12/9/19, 13/9/19	Inheritance	Inheritance Basics, Using super, Creating a Multilevel Hierarchy, When Constructors Are Executed, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, The Object Class.  Practical sessions	T1:8 R1:9 O1:5
17/9/19, 19/9/19	Packages	Packages, Access Protection, Importing Packages  Practical sessions	T1:9:1-3 R1:13 O1:4

20/9/19	Interface	Interfaces, Default Interface Methods, Use static Methods in an Interface, Final Thoughts on Packages and Interfaces.	T1:9:4-7 R1:12 O1:10
		Practical sessions	
24/9/19, 26/9/19	Introduction to Exception Handling	Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java's Built-in Exceptions	T1:10:1-7 R1:14 O1:13
		Practical sessions	
27/9/19, 1/10/19	Exception Handling	Creating Your Own Exception Subclasses, Chained Exceptions, Three Recently Added Exception Features, Using Exceptions.	T1:10:8-11 R1:14 O1:13
		Practical sessions	
	Unit-V		
3/10/19, 4/10/19	Introduction to Threads	The Java Thread Model, The Main Thread, Creating a Thread	T1:11:1-3 R1:20:1
		Practical sessions	
10/10/19, 11/10/19, 15/10/19	Multithreaded Programming	Creating Multiple Threads, Using isAlive ( ) and join ( ), Thread Priorities, Synchronization, Interthread Communication, Suspending, Resuming, and Stopping Threads. Obtaining A Thread's State, Using Multithreading.	T1:11:4-11 R1:20 O1:15
		Practical sessions	
17/10/19, 18/10/19, 22/10/19	Enumerations, Autoboxing, and Annotations (Metadata)	Enumerations, Type Wrappers, Autoboxing, Annotations (Metadata), Type Annotations, Repeating Annotations.	T1:12
		Practical sessions	
24/10/19 & 25/10/19	Revision part		



**X. Mapping course outcomes leading to the achievement of the program outcomes:**

Course Outcomes	Program Outcomes												
	a	B	c	d	e	f	g	H	i	j	k	l	m
1			S		H							S	
2	S				S								
3		S		H									S

**S = Supportive**

**H = Highly Related**

**Justification of Course syllabus covering Course Outcomes:**

By covering the syllabus a student can understand the designing of algorithm and flowcharts. Student is able to develop applications using C Program Constructs.

**Justification of CO's –PO's Mapping Table:**

By mapping CO-1 to the PO's C, E & L which are related to the course CO1: The student is able to analyze and Implement Problems

By mapping CO-2 to the PO's A, E, which are related to the course CO2: The student is able to efficient programs

By mapping CO-3 to the PO's B & D which are related to the course CO3: The student is able to create user-friendly interfaces and get the result.