

Course Title		Object Oriented Programming with JAVA	
Course Code		CE306	
Course Credit	Theory	:	03
	Practical	:	01
	Tutorial	:	00
	Credits	:	04
Course Learning Outcomes			
<p>On the completion of the course, students will be able to :</p> <ul style="list-style-type: none"> <li>• <b>Demonstrate</b> Object Oriented Programming Concepts in JAVA.</li> <li>• <b>Identify</b> properties and methods of entity and also able to design class.</li> <li>• <b>Develop</b> error free application.</li> <li>• <b>Create</b> packages (reusable code).</li> <li>• <b>Design</b> attractive user interface using AWT / Swing.</li> <li>• <b>Apply</b> parallel computations in solutions.</li> <li>• <b>Develop</b> programs to solve real life problems.</li> </ul>			
Detailed Syllabus			
Sr. No.	Name of chapter & details		Hours Allotted
<b>Section – I</b>			
1	<b>Introduction to JAVA</b> Overview of Object Oriented Programming Concepts, Features of Java Language, Architecture, Introduction to JDK, JVM, JRE, Structure of Java Program, Comments, Data types, Variables, Operators, Control Statements, Type conversion, String manipulation, Annotation, Enumeration, Wrapper class.		<b>08</b>
2	<b>Class and Object</b> Defining a Class, Creating Objects, Accessing Class Members, Constructors, Methods Overloading, Static Members, Nesting of Methods, Collections(List, Vector, and Array List).		<b>06</b>
3	<b>Inheritance and Interfaces</b> Types of Inheritance, Create a sub Class, Subclass constructor, Overriding, Final, finalize, visibility of field, inner class, Create Interfaces, Implement Interfaces, Access Interface Variables.		<b>08</b>
4	<b>Packages</b> Introduction, System Packages, Creating Packages, Accessing Packages.		<b>02</b>

**Section – II**

<b>5</b>	<b>Managing Errors and Exceptions</b> Types of errors, Exceptions, Syntax of Exception Handling Code, Multiple Catch Statements, Using finally Statement, Throwing Our Own Exceptions, Using Exceptions for Debugging.	<b>04</b>
<b>6</b>	<b>Multithreading</b> Introduction, Java Thread Model, Creating Threads, Extend Thread class, Using Thread Methods, Thread Exceptions, Thread Priority.	<b>06</b>
<b>7</b>	<b>Input/output and File handling</b> I/O Streams, File Class, byte stream, File Handling, Filtered byte streams, Object Streams, Character Stream, Random Access File.	<b>04</b>
<b>8</b>	<b>GUI in Java</b> Introduction to AWT, Basic of Swing, Introduction to Applet, JApplet class, Lifecycle of Applet, Applet HTML tag and Attributes, 2D shapes, Color and Font class, Swing Components – “JButton, JLabel, JTextArea, JRadioButton, JFrame, JPanel”	<b>06</b>
<b>9</b>	<b>Event Handling</b> Event Delegation Model, Event classes, The sources of events, The Event Listener Interfaces, Adapter classes.	<b>04</b>

**Instructional Method and Pedagogy**

- Lectures will be conducted in audio-visual class room to discuss important concepts with the help of animations / videos / PPTs / case studies to understand the concepts effectively.
- Problems based on concepts learnt in each unit/topic will be given followed by discussion to improve problem solving skills.
- Existing Application that used OOP concepts will be demonstrated to understand the use of concepts in real life application.
- Team project will be given such that students can apply their OOP skills.

### Reference books

- Hari Mohan Pande ,JAVA Programming,Pearson Education
- Herbert Schild, The Complete Reference Java 2, Fifth Edition, TMH
- Horstmann& Cornell, Core Java Volume – I& II, Eight Edition, Pearson Education
- Ivor Horton, Beginning Java 2 , Fifth Edition, Wrox Publication
- Horstmann, Big Java, Third Edition, Wiley-India
- John Zukowski, Java 2 J2SE 1.4, 2002, BPB Publication
- Katy Sierra & Bert Bates, Head First Java, Second Edition, O'Reilly Publication
- E Balagurusamy, Programming with Java, Second Edition, Tata McGraw Hill
- David Herst, Learning Java, First Edition, BPB Publication
- P Radhakrishna, Object Oriented Programming through Java, First Edition, Universities Press
- C Muthu, Programming with Java, First Edition, Thomson Publication

### Additional Resources

- [www.Java2s.com](http://www.Java2s.com)
- [www.roseindia.net](http://www.roseindia.net)
- [www.javabeginner.com](http://www.javabeginner.com)
- [docs.oracle.com/javase/tutorial/](http://docs.oracle.com/javase/tutorial/)
- [www.javatutorialhub.com/](http://www.javatutorialhub.com/)
- [Spoken-tutorials.org](http://Spoken-tutorials.org)

## List of Experiments

- 1 **Identify** latest version of JDK, Installation of JDK, GUI tools, Settings of environment variables (PATH, CLASSPATH). Also **write** JAVA program to display “Hello World”.
- 2 **Write** JAVA programs to **perform** the following computations:
  - 1)  $SQRT(SQRT(A) + B^{SQRT(A)})$
  - 2)  $\sin^{-1}X$
  - 3)  $Y^{\log x}$
  - 4)  $A \cot(X)$
  - 5)  $SQRT (ABS(a-b))$
- 3
  - 1) **Develop** a JAVA program to calculate income tax to be paid by an individual. Accept total taxable income using command line argument.
  - 2) **Write** a JAVA program to implement an advance calculator which performs basic arithmetic, logarithmic and trigonometric functions.
  - 3) **Write** a JAVA program to count odd number and even numbers from N value supplied in command line argument.
  - 4) **Write** a JAVA program to find minimum and maximum number from three numbers using single elseif -ladder.
- 4
  - 1) **Develop** a JAVA program to check whether given number is prime or not.
  - 2) **Write** a JAVA program to find whether given number is armstrong or not
  - 3) **Develop** a JAVA program to generate fibonacci series.
  - 4) **Develop** a JAVA program to reverse digits of a given number.
  - 5) **Develop** a Java program which takes three number from user as command line argument, “start”, “end” and “step”. Display all the numbers between “start” and “end” at given “step” interval.
- 5
  - 1) **Develop** a JAVA program to add first 20 prime numbers.
  - 2) **Produce** the following output using nested loops
 

```

1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
          
```
  - 3) Develop a JAVA program to calculate  $1!+3!+5!+\dots+n!$ . Value of n should be given by user.
  - 4) **Develop** a JAVA method to find all primes for a given range.

- 6** Write the following programs based on Array:
- 1) Compute average of N numbers.
  - 2) Sort all elements in ascending order.
  - 3) Search whether given number exist in array or not.
  - 4) Sort array of string in alphabetical order.
- 7** Write JAVA programs for the followings:
- 1) **Perform** addition of two matrices of dimension M x N.
  - 2) **Compute** Multiplication of two matrices A & B of dimension M x N and P x Q respectively.
  - 3) **Recognize** highest from each row in 2D array of size M x N.
  - 4) **Apply** sorting method on each row of Jagged Array.
- 8**
- 1) **Develop** a JAVA program which accepts two strings, "str1" and "str2" using DataInputStream class. **Check** whether "str1" contains "str2" or not. Display appropriate message.
  - 2) **Develop** a JAVA program which accept a String "str1" from user and also perform string operations (uppercase , lowercase, trim, length, reverse, check for palindrome, substring, indexof) using textual menu.
  - 3) **Compute** the frequency of a word in given string.
- 9**
- 1) **Design** a class "Rectangle". **Identify** all the properties and methods of the class. Also **write** an appropriate constructor for the same. **Create** an Object of the class in JAVA program and demonstrate all the methods.
  - 2) **Design** a class "Time" having instance variable "hour", "minute" and "second". **Write** appropriate constructor for the same. Write getters and setters for all the instance variables. Also write method to convert time object in seconds. Use the Time class in JAVA program.
  - 3) **Design** a class "Person". **Identify** instance variable with appropriate data types. **Create** constructor for the same. Use the class in JAVA program.
- 10**
- 1) **Modify** the "Rectangle" class designed in Tutorial 9 to perform constructor overloading. Use this class in JAVA program.
  - 2) **Modify** the "Person" class designed in Tutorial 9 9 to perform constructor overloading. Use this class in JAVA program.
  - 3) **Design** a "Student" class with appropriate instance variable (static). Use the student class in JAVA program and calculate the number of objects created. Also sort the Students in acceding order of their name.
- 11**
- 1) **Create** a JAVA program to find sum of two, three, four or N numbers using method overloading or variable arguments concept.
  - 2) **Create** a JAVA program to find maximum of two, three, four or N

numbers using method overloading or variable arguments concept.

- 12**      1) **Create** a package “rku” that contains classes “Rect” & “Time”. Add one more class “Box” as extension of “Rect” class that override few methods of parent class. **Analyze** the instance variable and necessary methods of each class. Also integrate this package to create JAVA programs for the followings:
1. to **perform** find largest rectangle among N rectangles using array of objects.
  2. to find volume of N boxes.
- 13,14**    1) **Design** a class “Account”. Identify all the instance variable. Extend this class to “SavingAccount” and “CurrentAccount” to make them more specific. Include necessary methods in order to achieve following tasks.
- Accept deposit from a customer and update the balance.
  - Display the current balance.
  - Compute and deposit interest.
  - Permit withdrawal and update the balance.
  - Check for the minimum balance, impose penalty if necessary.
- 15**      1) **Design** an Interface called “RemoteControl”. Specify the functionality provided by remote control like “next”, “previous”, “stop”, “play” etc. Design classes “TV”, “DVD” and “FM”. Write methods of RemoteControl to implement the functionality. Demonstrate it Using JAVA program.
- 16**      1) **Write** a Java program to **demonstrate** the use of final keyword.  
2) **Write** a Java program to **demonstrate** the use of finalize method.
- 17**      1) **Write** a JAVA program to **compute** sum of 1 to 100 using 5 threads. Where each threads compute sum of 20 numbers.  
2) **Transform** the above problem to compute sum of 1 to N numbers using 5 threads.  
3) **Write** a JAVA program to solve producer-consumer problem using thread synchronization.
- 18**      1) **Create** a JAVA program that takes filename and search word as an input from command line arguments and checks the followings
- Whether that file exists or not.
  - If exists, display all lines of file that contains given search word.
  - If doesn't exist, print appropriate message.
- 2) **Create** a .csv file “rku.csv” that contains “roll\_number”, “name” and “branch”. Write a JAVA program to manipulate (CRUD) this file.

- 19**            1) **Create** JAVA applet to print “Hello” message on output screen.  
                  2) **Write** a JAVA program to demonstrate Applet Life Cycle.
- 20**            1) **Create** Applet code, save it in drive D:/JAVA .**Design** a web page (html) using CODEBASE attribute and save it in E:/MYJAVA. Execute same code using Appletviewer.
- 21**            1) **Design** JAVA applet program that allows user to draw lines, rectangles and ovals.  
                  2) **Design** a web page to accept “name” and “address” from user. Write applet code to receive these inputs using parameter and display on the screen.
- 22, 23**        1) **Write** applet code to draw smiley face using graphics class.  
                  2) **Develop** a JAVA Applet to design login form.
- 24,25**        1) **Design** an Applet that create two buttons named “Red” and “Blue” when a button is pressed the background color of the applets is set to the color named by the button’s label.  
                  2) **Write** a Program to create a List Box and a Text Area. Fill up the List Box with some file names. When user double clicks on any filename of the list box, the file should be opened and its contents should be displayed in the text Area.
- 26,27**        1) **Develop** JAVA application “calculator” using panel.

**Note:** Following features should be accommodated in application.

- a) Background color of each button should be changed on focus. And reset to original color on blur.
  - b) At least implement five features of scientific calculator.
- 28**            1)            **Design** JAVA Applet to Scroll image using Scrollpane.