`

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

DUNDIGAL, HYDERABAD -500 043

COMPUTER SCIENCE AND ENGINEERING

JAVA LIST OF EXPERIMENTS

Course Title	JAVA PROGRAMMING LAB			
Course Code	A40585			
Regulation	R13 – JNTUH			
Course Structure	Lectures	Tutorials	Practicals	Credits
	--	--	3	2
Course Coordinator	Mr. P Ravinder, Assistant Professor			
Team of Instructors	Ms .G Geetha, Assistant Professor			

I. COURSE OVERVIEW:

This course explains the fundamental ideas behind the object oriented approach to programming. Knowledge of java helps to create the latest innovations in programming. Like the successful computer languages that came before, java is the blend of the best elements of its rich heritage combined with the innovative concepts required by its unique environment. This course involves OOP concepts, java basics, inheritance, polymorphism, interfaces, inner classes, packages, Exception handling, multithreading, collection framework, files, JDBC and GUI components. This course is presented to students by power point projections, course handouts, lecture notes, course handouts, assignments, objective and subjective tests.

II. COURSE OBJECTIVES:

At the end of the course, the students will be able to:

I. **Understand** object oriented programming concepts and apply them in problem solving.
II. **Understand** the basics of java Console and GUI based programming.
III. **Demonstrate** how the multi tasking is performed by using threads and types of exception handling.
IV. **Describe** the byte streams and character streams for file management and relate java applications to database for connectivity.

III. COURSE OUTCOMES:

After completing this course the student must demonstrate the knowledge and ability to:

1. **List** and use Object Oriented Programming concepts for problem solving.

2. **Apply** java programming on different operating systems to understand portability.

3. **Solve** the inter-disciplinary applications using the concept of inheritance.

4. **Develop** programs using Java collection API as well as the java standard class library.

5. **Produce** design stability for various applications- by applying exception handling and inheritance.

6. **Use** multithreading to reduce the wastage of CPU time.

7. **Construct** JDBC to provide a program level interface for communicating with database using java programming.

`

8. **Create** java programs using console and GUI based programming.
9. **Simulate** dynamic and interactive programs using applets.
10. **Use** of object oriented programming is easy to communicate between developers and users.

IV.   LIST OF EXPERIMENTS:

Division of Experiments	List of Experiments
Basics of java	**Week – 1** 1. Capitalize the first letter in A and B and print them on a single line, separated by a space. 2. Swap numbers without using third variable. 3. Given a square matrix of size N×N, Design a class to calculate the absolute difference between the sums of its diagonals. 4. Design a class for complex numbers in Java. In addition to methods for basic operations on complex numbers, provide a method to return the number of active objects created. 5. Read a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of java.util). 6. Read array of numbers through command line and sort in ascending order.
Methods Constructors, Inheritance	**Week – 2** 1. Design a class to represent a bank account. Which include contains account number, name of the depositor, type of the account, balance amount in the account. Define Methods, to assign initial values, to Deposit an amount, to Withdraw amount after checking balance, to display name and balance. 2. Modify the above program to incorporate a constructor to provide initial values. 3. Assume that a bank maintains two accounts for its costumer's one called savings and other current account. The savings account provides compound interest and withdrawal facility but no check book facilities. The current account provides check book facility but no interest. Implement this application using inheritance concept. 4. Create a class Account that stores customer name,account number and type of account. from this derive the classes Curr_act and Sav_acct to make them more specific to their requirements.Include the necessary methods to achieve the following tasks • Accept deposit from a customer and update the balance • Display the balance • Permit withdrawal and update the balance Include constructors for all three classes. 5. Create an inheritance hierarchy of Rodent, Mouse, Gerbil, Hamster etc. In the base class provide methods that are common to all Rodents and override these in the derived classes to perform different behaviors, depending on the specific type of Rodent. Create an array of Rodent, fill it with

		different specific types of Rodents and call your base class methods. 6. Design a class to represent a Student details. include the Student ID, name of the Student, branch, year and assign initial values, calculate average of marks of 6 subjects, calculate attendance percentage.

Polymorphism, Interfaces		**Week – 3**
		1. Design a Java application to create an abstract class named Shape that contains two integers and an empty method named print Area().provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
		2. Develop with suitable hierarchy, classes for Point, Shape, Rectangle, Square, Circle, Ellipse, Triangle, Polygon, etc. Design a simple test application to demonstrate dynamic polymorphism.
		3. Creates a class called FixedStack that implements a fixed-length version of an integer stack.
		4. Write a program that shows the partial implementation of Interface.
		5. Your teacher has given you the task of drawing a staircase structure. Being an expert programmer, you decided to make a program to draw it for you instead. Given the required height, can you print a staircase as shown in the example
		
		           #
		       # #
		   # # #
		# # # #
		
		6. When a subclass inherits a superclass, it can override methods of the superclass.Considerthefollowing
		Sports class:
		Class Sports
		{
		  String get_name()
		    {
		      Return"generic Sports";
		    }
		  Void get_number_of_team_members()
		   {
		
		      System.out.println("eachteamhasnplayersin"+get _name());
		   }
		}
		   Now we want to create a *Soccer* class that inherits the *Sports* class. We can override the*get_name* method and return a different string.

Exception handling		**Week – 4** 1. You will be given two integers x and y as input, you have to compute x/y.Implement a class which raise an exception if x and y are not 32 bit signed integers or if y is zero. 2. Create a class Simple calculator .Use a grid layout to arrange buttons for the digits and for the +,-,*,% operations. Add a text field to display the result. Handle any possible exceptions like dividend by zero.

	3. Create a class MyCalculator which consists of a single method power(int,int). This method takes two integers, n and p, as parameters and finds n^p. If either n or p is negative, then the method must throw an exception which says "n and p should be non-negative". 4. Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box. 5. Define an exception called " NoMatchException" that is thrown when a string is not equal to "India". Write a program that uses this Exception. 6. Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the textfields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException Display the exception in a message dialog box.
Multithreading	**Week – 5** 1. Write a program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number. 2. Write a program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds. 3. Write a program to create a child process and allow the parent to display "parent" and the child to display "child" on the screen 4. Write a program to create a Zombie process. 5. Write a program that illustrates how an orphan is created. 6. Write a multi-threaded Java program to print all numbers below. 7. 100,000 that are both prime and fibonacci number (some examples are 2, 3, 5, 13, etc.). Design a thread that generates prime numbers below 100,000 and writes them into a pipe. Design another thread that generates fibonacci numbers and writes them to another pipe. The main thread should read both the pipes to identify numbers common to both. 8. Write a java program that correctly implements producer-consumer problem using the concept of Inter Thread communication.
Files, Layouts	**Week – 6** 1. Write a program that reads a file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of

	file and the length of the file in bytes.
	2. Write a program to implement character streams (reader classes).
	3. Write a program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +,-,*, % operations. Add a text field to display the result. Handle any possible exception like divided by zero.
	4. Suppose that table named table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using labels in grid layout.
	5. Write a Program to implement the flow layout and border Layout.
	6. Write a program to skip n bytes while reading the file using skip method of Java FileInputStream class.
Event Handling	**Week-7** 1. Simulates a traffic light Application. The Application lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button an appropriate message with "STOP" or "READY" or "GO" should appear above the buttons in selected color. Initially, there is no message shown. 2. Write a program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired. 3. Write a program to demonstrate the key event handlers. 4. Write a java program that allows the user to draw aline,rectangles and circle. 5. Write a Program to Demonstrate System Clock. 6. Write a program on event handling using Inner Classes.

Applets	**Week-8**
	 1. Develop an applet to print Chessboard Pattern.
	 2. Develop an applet that receives an integer in one text field and computes its factorial value and returns it in another text field, when the button named "compute" is clicked.
	 3. Develop an applet that receives three numeric values as input from the user and then displays the largest of the three on the screen. Write a HTML page and test the Applet.
	 4. Develop an applet that implements a simple client/server application. The client sends data to a server. The server receives the data, uses it to produce a result and then sends the result back to the client. The client displays the result on the console. For ex: The data sent from the client is the radius of a circle and the result produced by the server is the area of the circle.
	 5. Develop an applet do the following:
	   i. To output the question "Who is the inventor of Java?"
	   ii. To accept an answer.
	   iii. To printout "GOOD" and then stop if the answer is correct.
	   iv. To output the message "TRY AGAIN", if the answer is wrong.
	   v. To display the correct answer, when the answer is wrong even at the third attempt.
	 6. Write an Applet that computes the payment of a loan on the amount of the loan,the interest rate and the number of

	months.It takes one parameter from the browser. Monthly; if true, the interest rate is per month, otherwise the interest rate is annual.
JDBC Connectivity , Files	**Week – 9** 1. Write a Program that connects to a database using JDBC and does add, delete, modify and retrieve operations. 2. Write a Java Program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab(/t).It takes a name or phone number as input and prints the corresponding other value from the hash table.(hint use hash tables). 3. Implement the above program with database instead of a text file. 4. Write a Java Program that takes tab separated data (one record per line) from a text file and insert them into a database. 5. Write a Java Program that prints the metadata of a given table. 6. Create TYPE 1 driver program for JDBC with Access.
SWING	**Week-10** 1. Create a label with image icon and text using Java Swing JLabel class. 2. Write a Program to get X and Y coordinates using Java Swing JLabel class. 3. Write a java program to set background and foreground color of JLabel using Java Swing JLabel class. 4. Write a Program to set and get vertical alignment and Horizontal alignment of label's content using Java Swing JLabel class. 5. Write a program to set background color of JtextField using Java Swing JTextField class. **6.** Write a program to set foreground color of JtextField using Java Swing JTextField class.
AWT	**Week-11** 1. Write a program to change button background color using AWT Button class. 2. Write a program to determine if the button is enabled or not using isEnabled method. 3. Write a program Remove AWT Checkbox From Applet or Window Frame class 4. Write a program to remove a Checkbox from window frame or an applet. 5. Write a program to select a particular item of a choice or a combo box control using select method of AWT Choice class. **6.** Write a program to determine if the specific index is selected from a list using Java AWT List class.

Prepared by: Mr. P Ravinder, Assistant Professor
                Ms. G Geetha, Assistant Professor

HOD, CSE