

CODE: CSE212

CSE 212 : JAVA PROGRAMMING LAB

IV Sem BE (CS&E)

(2013)

DEPT OF COMPUTER SCIENCE & ENGG.

M. I. T., MANIPAL

Prepared By :

Dr. Harish S. V.
Mr. Chidananda Acharya
Ms. Roopashri Shetty

Approved by :

H.O.D., CSE DEPT.

INSTRUCTIONS TO STUDENTS

1. Students should be regular and come prepared for the lab practice.
2. In case a student misses a class, it is his/her responsibility to complete that missed experiment(s).
3. Students should bring the observation book, lab journal and lab manual. Prescribed textbook and class notes can be kept ready for reference if required.
4. They should implement the given experiment individually.
5. While conducting the experiments students should see that their programs would meet the following criteria:
 - 1 Programs should be interactive with appropriate prompt messages, error messages if any, and descriptive messages for outputs.
 - 2 Programs should perform input validation (Data type, range error, etc.) and give appropriate error messages and suggest corrective actions.
 - 3 Comments should be used to give the statement of the problem and every function should indicate the purpose of the function, inputs and outputs
 - 4 Statements within the program should be properly indented
 - 5 Use meaningful names for variables and functions.
 - 6 Make use of Constants and type definitions wherever needed.
 - 7 Once the experiment(s) get executed, they should show the program and results to the instructors and copy the same in their observation book.
 - 8 Questions for lab tests and exams need not necessarily be limited to the questions in the manual, but could involve some variations and / or combinations of the questions.

CONTENTS

<u>SL NO.</u>	<u>TITLE OF EXPERIMENT</u>	<u>PAGE NO.</u>
1.	Control Statements, Arrays and Matrices	5
2.	Classes and Methods	5
3.	Stacks and Lists	6
4.	Strings	6
5.	Inheritance and Packages	7
6.	Interfaces and Exception Handling	8
7.	Threads and Input / Output	8
8.	Threads and Input / Output	8
9.	Applets and Event Handling	9
10.	AWT and Swings	9
11.	AWT and Swings	9
12.	JDBC	10

PROCEDURE OF EVALUATION

There will be continuous evaluation which is for 60 Marks. End exam will be of 40Marks.

Implementation of experiments, Observation and / or Journal and Viva Voce	60% (60 Marks)
Test	40% (40 Marks)

Week 1: Control Statements, Arrays and Matrices

Write java programs for the following. (Inputs may be given directly in the program and not through the keyboard)

1. To find the sum of digits of a number (assume 4 digits).
2. To find the factorial of the given number using iteration.
3. To generate the first n fibonacci numbers using recursion. The recursive method is in the same class as main.
4. Create and initialize an integer array. Search for particular element in the array using linear search.
5. Create and initialize 2 matrices. Find their sum. Display all 3 matrices.

Week 2: Classes and Methods

1. Define a class to represent a complex number called **Complex**. Provide the following member functions:-
 - i. To assign initial values to the Complex object.
 - ii. To display a complex number in a+ib format.
 - iii. To add 2 complex numbers. (the return value should be complex)
 - iv. To subtract 2 complex numbers
 Write a main function to test the class.
2. Create a class called **Time** that has data members to represent hours, minutes and seconds. Provide the following member functions:-
 - i. To assign initial values to the Time object.
 - ii. To display a Time object in the form of hh:mm:ss {0 to 24 hours}
 - iii. To add 2 Time objects (the return value should be a Time object)
 - iv. To subtract 2 Time objects (the return value should be a Time object)
 - v. To compare 2 time objects and to determine if they are equal or if the first is greater or smaller than the second one.
- 3a. Define a class named **Movie**. Include private fields for the title, year, and name of the director. Include three public functions with the prototypes


```
void Movie::setTitle(char [ ]);
void Movie::setYear(int);
void Movie::setDirector(char [ ]);
```

 Include another function that displays all the information about a Movie.
- 3b. Include a function which accepts 2 objects of type Movie and displays whether or not they were released in the same year and also whether the Directors are same. String functions may be used.
4. Create a class called **Stack** for storing integers. The data members are an integer array for storing the integers and an integer for storing the top of stack (tos). Include member functions for initializing tos to 0, pushing an element to the stack and for popping an element from the stack. The push() function should check for “stack overflow” and pop() should check for “stack underflow”.

Week 3: Stacks and Lists

1. Design a class, which represents a stack of integers. Provide constructors as well as methods namely push(), pop(), displayFromTop(), displayFromBot(), stackTop(), isEmpty() and isFull() which respectively pushes an integer onto the stack, pops an integer from the stack, displays the stack contents from top to bottom, displays the stack contents from bottom to top, displays the top element of the stack, determines whether the stack is empty or not and determines whether the stack is full or not.
2. Design a stack class and provide all the previously listed methods so that it works with a stack of student records. Every student record should have a name and id field.
3. Design a class, which represents a list of integers. Provide a default constructor to this class. Also provide methods namely:- isEmpty(), Display(), AddAtLast() and DeleteAtHead() Write a main method, which uses these methods.
4. Modify the above so that the list works with student records, where every record has a name and id field.
5. Create two linked lists with elements (1,2,7,8) & (3,4,5,6) respectively. Merge the two lists into one linked list. The merged list must be sorted one as well, i.e., it should contain the elements (1,2,3,4,5,6,7,8)
6. Implement the stack operations using a linked list.

Week 4: Strings

NOTE:- *Implement all parts as separate programs.*

- 1a) Design a class which represents a student. Every student record is made up of the following fields.
 - i) Registration number (int)
 - ii) Full Name (String)
 - iii) Date of joining (Gregorian calendar)
 - iv) Semester (short)
 - v) GPA (float)
 - vi) CGPA (float)

Test the class by writing suitable main method.
- b) Provide default, copy and parameterized constructor to this class. Write display method which displays the record. Test it in main.
- c) Whenever a student joins he will be given a new registration number. Registration number is calculated as follows. If year of joining is 2012 and he is the 80th student to join then his registration number will be 1280.
- d) Create an array of student record and store minimum of 5 records in it. Input the records and display them.
- e) Sort the student records with respect to semester and CGPA.
- f) Sort the student record with respect to name.

- g) List all the students whose name starts with 'S'
- h) List all the students name whose name contains "Naik".
- i) Change the full name in the object to name with just initials and family name. For example, Prakash Kalingrao Aithal must be changed to P.K. Aithal and store it in the object. Display modified objects.

Week 5: Inheritance and Packages

1. Design a class, which represents a patient. Fields of this class are name, age and hospital number. Provide methods to input and display all fields. Next design a class called Inpatient, which inherits from patient class. Provide fields to represent department name, admission date and room type. Provide methods to input and display these fields. Next design a class called Billing, which inherits from Inpatient class. Provide a field to represent the discharge date. Provide methods to input this field value as well as to display the total amount. The total amount is calculated based on room type and doctor charges as shown below:-

RoomType	Consultancy Charges/day	Room Rent Charges/day
Special	Rs. 1000.00	Rs. 200
SemiSpecial	Rs. 500.00	Rs. 100
General	Rs. 100.00	Rs. 50

Show the use of a Billing object in main (Make use of super keyword).

2. Develop following methods
 - (i) max (which finds maximum among three integers and returns the maximum integer)
 - (ii) max (which finds maximum among three floating point numbers and returns the maximum among them)
 - (iii) max (which finds the maximum in an array and returns it)
 - (iv) max (which finds the maximum in a matrix and returns the result)

Place this in a package called p1. Let this package be present in a folder called "myPackages", which is a folder in your present working directory (eg:- c:\student\4rthcse01\mypackages\p1). Write a main method to use the methods of package p1.
3. Create an abstract class Figure with abstract method area and two integer dimensions. Extend this class to inherit three more classes Rectangle, Triangle and Square which implements the area method. Show how the area can be computed dynamically during run time for Rectangle, Square and Triangle.
4. Design a class, which represents a double-ended Stack (Stack operations can be performed on both ends of the Stack). Let this class inherit from the stack class. Provide all other additional methods to this class. Show the usage of this class in a main method.

Week 6: Interfaces and Exception Handling.

1. Design an interface called Stack with 2 methods namely push() and pop() in it. Design a class called FixedStack, which implements a fixed length version of the stack. Also design a class called DynamicStack, which implements a growable version of the stack. Write a main method, which uses both these classes through an interface reference.
2. Design a class, which represents an employee. The data members are:- name (string), age (int), grossSalary (double), takeHomeSalary (float), grade (char). Provide methods called input() and display() which reads all details of a record from the keyboard and displays them respectively. Handle IOException while reading from the keyboard. Provide a menu with the options: Input, Display and Exit to read users choice. (Make use of Wrapper classes)
3. Design a Student class. Provide your own exceptions namely Seats Filled exception, the exception is thrown when Student registration number is >XX25 (where XX is last two digits of the year of joining) Show the usage of this exception handling in Student object in the main.(Note: Registration number must be generated as in 4th lab question number 1c.)
4. Design a stack class. Provide your own stack exceptions namely push exception and pop exception, which throw exceptions when the stack is full and when the stack is empty respectively. Show the usage of these exceptions in handling a stack object in the main.

Week 7 & 8: Threads and Input / Output

1. Write a java program to create and initialize a matrix of integers. Create n threads where n is equal to the number of rows in the matrix. Each of these threads should compute a distinct row sum. The main thread computes the complete sum by looking into the partial sums given by the threads.
2. Create a class StudentThread. Create n threads in it (n=total number of semesters). Each thread finds the Student with the highest CGPA in a particular semester. The main thread finds the student with the highest CGPA among all semesters.
3. Write information of n employees to a file. This information should be read from the keyboard. The data members of employee class are:- name (string), age (int), GrossSalary (double), TakeHomeSalary (float), Grade (char). (Each record can be stored on a separate line with a blank separating every 2 fields) Transfer the records with grade 'A' to another file. Also display those records on the screen.
4. Write a program to display the listing of a given directory. Recursion can be used.
5. Assume that there exists a file with different file names stored in it. Every file name is placed on a separate line. Create two threads, which scan the

- first half, and the second half of the file respectively, to search the filenames which end with .cpp. Write those file names onto the screen.
6. Count the number of single characters, numbers (sequence of 1 or more digits) , words and lines from a file Placed.txt and place the result on the screen.
 7. Write a program to copy one file to other using
 - i)char stream classes ii)Byte stream classes
 8. Write a program which reverses the lines in the file using linked list.

Week 9: Applets and Event Handling

1. Create a HTML file which contains your bio-data Write an applet which transfers the control to your bio-data when mouse is clicked. Make use of mouse adapter and anonymous inner class concept. Note:-You have to run the Applet using browser Applet Viewer is not capable of running this applet.
2. Develop an applet which gets the message from User via Key Board, and displays the message at the center of the Applet when enter is pressed. Applet must scroll the message horizontally from left to right when left arrow key is pressed and it must scroll the message horizontally from right to left when right arrow key is pressed. Entire message must move up or down depending on whether up arrow or down arrow is pressed. When any other key is pressed it should be shown in the status bar. Set the background and foreground colors of the banner to cyan and red respectively.
3. Write an applet that obtains an integer from the user and displays the multiplication table for that number.

Week 10 & 11: AWT and Swings

1. Write Java programs to demonstrate the usage of following AWT controls: Buttons, Check Boxes, Checkbox Group, Combo Boxes, Radio Buttons, Lists, Text Field, Text Area, Tabbed Panes, Scroll Panes.
2. Write a Java program to demonstrate the usage of Menu Bar and Menus. Use Dialog Boxes to display the Menu options selected.
3. Write Java program that allows the user to draw a rectangle by dragging the mouse on the application window. The upper-left coordinate should be the location where the user presses the mouse button, and the lower-right coordinate should be the location where the user releases the mouse button. Also display the area of the rectangle.
4. Modify the above program to draw a shape with the mouse. The shape should be allowed to choose from an oval, an arc, a line, a rectangle with rounded corners and predefined polygon. Also display the mouse coordinates in the status bar.

5. Write a Java program that displays a Circle of random size and calculates and displays the area, radius, diameter and circumference.
6. Enhance the above program by allowing the user to alter the radius with a scrollbar. The program should work for all radii in the range 100 to 200. As the radius changes, the diameter, area and circumference should be updated and displayed. The initial radius should be 150.
7. Write a Java program to simulate a static analog clock whose display is controlled by a user controlled static digital clock.
8. Write a Java program to implement a simple calculator.
9. Write an applet to test username and password.
10. Write an applet that obtains two floating point numbers from the user and displays the sum, product, difference and quotient of these numbers.

Week 12 : JDBC

1. Create a simple student table that contains regno, name and cgpa. Write a Java application which will populate the student table. Make use of AWT controls to get the data from the user.
2. Write a Java program which runs the user supplied query on student database. Display the result of the query onto the text area.

Text and References

1. Herbert Schildt , “The Complete Reference Java ”,Seventh Edition
2. Steven Holzner, “Java 2 programming Black Book” , (JDK 5 Edition), DreamTech, New Delhi
3. Art Gittleman, “Ultimate Java Programming”, Second Edition, dreamtech
