II B. Tech – II Semester

(20ES4109) OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

Int. Marks	Ext. Marks	Total Marks		L	T	P	C
15	35	50		-	-	3	1.5

Pre-Requisites:

Course Objectives:
This course is designed to:
- To understand object oriented concept
- To become familiar with classes and objects
- Understand various kinds of methods
- Introduce built-in and user defined exceptions
- To become familiar the concepts of AWT

Course Outcomes:
At the end of the course the student will be able to
1. Create simple applications using classes, objects and inheritance
2. Apply parallel processing applications using threads
3. Develop applications GUI applications using AWT

Exercise - 1 (Basics)
a) Write a JAVA program to display default value of all primitive data type of JAVA.
b) Write a case study on public static void main (250 words).
c) Five Bikers Compete in a race such that they drive at a constant speed which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all 5 racers. Take as input the speed of each racer and print back the speed of qualifying racers.

Exercise - 2 (Control -flow)
a) Write a JAVA program to search for an element in a given list of elements using binary search mechanism.
b) Write a JAVA program to sort for an element in a given list of elements using bubble sort

Exercise - 3 (Class, Objects)
a) Define a class called CalAge. This class is used to calculate age of a person from her or his date of birth and the current date. Include a mutator method that allows the user to enter her or his date of birth and set the value for current date. Also include a method to return the age in years and months (for example, 25.5 years) as a double value. Include an additional method to check if the date of birth entered by the user is a valid one. For example, 30 February 2008 is an invalid date. Embed your class in a test program.
b) Define a class called Journal that could be used to store an entry for a research paper that will be published. The class should have instance variables to store the author's name, title of the paper, and the date of submission using the Date class from this chapter. Add a constructor to the class that allows the user of the class to set all instance variables. Also add a method, display details, that outputs all the instance variables, and another method called get Submission Details that returns the title of the paper, with the first letter of each word capitalized. Test your class from the main method.

Exercise - 4 (Static)

Define a class called Book Store to maintain the record of books sold. The store contains three categories of books i.e. "Kids", "Engineering", and "Story". The following details should be maintained for each book. Book category, Author, Title, Publisher, Selling price of the book, Quantity. Create a constructor that initializes the Store ID and the rest of the details of the book mentioned above. Include a method named track Sales Status that will display the total number of books sold by the store (add a static variable that tracks the total number of books sold). Also, include a method to display the quantity available corresponding to each Book ID. Write a main method to test your class.

Exercise - 5 (Methods)

a) You are running a courier agency. The weight of a parcel determines the number of stamps that will be needed to send that parcel. For each kilogram, a stamp of $2 is needed. Create a class to accept the weight of five parcels in floating-point values. Also, the courier company charges an additional rate depending on where the courier has to be delivered. The charges are $20 for delivery within the city of posting, and $40 for delivery anywhere else in the country. Write a computer program to calculate and display the total cost of each parcel depending on the weight and delivery location of the parcel.

b) Write a program that takes as input a set of 15 numbers from the keyboard into an array of type int[] . Create another array that will also read 15 other numbers of type int into it. Now merge the elements of these two arrays into one. The output is to be a two-column list. The first column is a list of the distinct array elements; the second column is the count of the number of occurrences of each element.

Exercise - 6 (Inheritance)

Define a class named Person that contains two instance variables of type String that stores the first name and last name of a person and appropriate accessor and mutator methods. Also create a method named display Details that outputs the details of a person. Next, define a class named Student that is derived from Person , the constructor for which should receive first name and last name from the class Student and also assigns values to student id, course, and teacher name. This class should redefine the display Details method to person details as well as details of a student. Include appropriate constructor(s). Define a class named Teacher that is derived from Person . This class should contain instance variables for the subject name and salary. Include appropriate constructor(s). Finally, redefine the display Details method to include all teacher information in the printout. Create a main method that creates at least two student objects and two teacher objects with different values and calls display Details for each.

Exercise - 7 (Exception)

a) Create a class named Employee that can be used to calculate the salaries of different employees. The Employee class should keep a track of the employee ID, name, department, salary, and designation with appropriate accessor and mutator methods. Also create an equals() method that overrides Object's equals() method, where employees can check if their designation is identical. Next, create two additional classes named Manager and Clerk that are derived from Employee. Create an overridden method name

b) Write a case study on public static void main (250 words).

c) Add Bonus that returns the salary of the employee after adding up the bonus. There is a default bonus of $200/month. Managers have a bonus of $300/month and clerks have a bonus of $100/month. Finally create a display method to print the details of the employee. You may assume the initial salary of an employee and other necessary values. Test your classes from a main method.

d) Extend the previous problem to calculate the salary deductions based on the number of days an employee is on leave. Consider 20 working days per month. Add a method that calculates the deductions of each employee based on their leave record. In your main method, create an array of type deduction filled with sample data of all types of Employees. Finally calculate the total deduction that iterates through the array and returns the total amount of deductions of all the employees in a month.

Exercise - 8 (Exception handling)

a) Define a class to maintain bank accounts of customers. The program should place the code into a try-catch block with multiple catches to check for the validity of various attributes based on the following criteria.

    1. Customer ID must start with a letter and should be followed by three digits.

    2. Account number must be of five digits.

    3. Initial balance must be above $1000.

    4. Print suitable error matches within the catch block. If any of the criteria mentioned above is not fulfilled, the program should loop back and let the user enter new data.

b) Modify the previous exercise to include methods for amount deposited and amount withdrawn. Create your own exception class which will check inside the method for the amount deposited so that after the deposit, the maximum balance in the account must not be more than $5000. Also, check inside the method for amount withdrawn so that the available balance after the withdrawal does not go below $1000. Invoke the defined methods from your main method and catch the exceptions.

Exercise - 9 (Threads)

a) Write a program to simulate a buzzer. The program should make use of the Thread class. Create a class named Buzzer that extends the Thread class. Create an interface named Monitor Time which contains a method setBuzzerTime() . Your Buzzer class should implement this interface. Override setBuzzerTime() method to set the buzzer time delay in milliseconds and the number of times the buzzer should be repeated. Also, include methods blow Buzzer to start the buzzer.

b) Write a Java program which handles Push operation and Pop operation on stack concurrently.

c) Write a Java program which first generates a set of random numbers and then determines negative, positive even, positive odd numbers concurrently.

Exercise - 10 (File Streams)

a) Write a program that will count the total occurrences of the number '10' in a text file of strings representing numbers of type int and will show the value of the count on the screen once the whole file is read. The file contains the following numbers separated by space.
10 4 7 8 10 34 11 10 15 6 10

b) Write a program that reads grades of type double of eight students that the user provides. The grades lie between 0 and 10. These grades should be written to a binary file and read from it. The program outputs the highest and lowest grades achieved by students on the screen. The file contains nothing but numbers of type double written to the file with write Double.

The Sieve of Erastothenes is an ancient algorithm that generates prime numbers. Consider the list of numbers from 2 to 10 as follows:

                        2 3 4 5 6 7 8 9 10

The algorithm starts with the first prime number in the list, which is 2, and then iterates through the remainder of the list, removing any number that is a multiple of 2 (in this case, 4, 6, 8, and 10), leaving

                        2 3 5 7 9

We then repeat the process with the second prime number in the list, which is 3, and then iterate through the remainder of the list, removing any number that is a multiple of 3 (in this case 9), leaving

                        2 3 5 7

We then repeat starting with each successive prime number, but no elements are removed because there are no multiples of 5 or 7 (a more efficient implementation of the algorithm would stop without examining 5 or 7). The numbers that remain in the list are all prime numbers. Implement this algorithm using an Array List of integers that is initialized to the values from 2 to 100. Your program

---

can iterate numerically through the Array List from index 0 to index size()-1 to get the current prime number, but should use an Iterator to scan through the remainder of the list to eliminate the multiples. You can use the list Iterator method to retrieve the iterator starting at a specified index into the Array List. Output all remaining prime numbers to the console.

Exercise - 11 (AWT)
a) Write a JAVA program to paint like paint brush in applet
b) Write a JAVA program that display the x and y position of the cursor movement using Mouse
c) Write a JAVA program to build a Calculator in AWT

Course Outcomes:

A student who successfully fulfills this course requirement will be able to:

S. No	Course Outcome	BTL
CO1	Able to use Java compiler and eclipse platform to write and execute java program.	L4
CO2	Understand and Apply Object oriented features and Java concepts.	L4
CO3	Able to apply the concept of multithreading and implement exception handling.	L4
CO4	Able to access data from a Database with java program.	L4

Correlation of Cos with POs & PSOs:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	0	1	1	1	1	0	0	2	2	3	2	2
CO2	2	2	0	1	1	1	1	0	0	2	2	2	2	2
CO3	2	2	0	1	1	1	1	0	0	2	2	3	3	2
CO4	2	2	0	1	1	1	1	0	0	2	2	3	3	2