2023-24

اور فن قوم کوئی جب
جاتی ہو عاری سے علم
کو غربت وہ تو ہے
اور ہے دیتی دعوت
تو ہے آتی غربت جب
کو جرائم ہزاروں وہ
ہے۔ دیتی جنم

"When a nation becomes devoid of art and learning, it invites poverty and when poverty comes it brings in its wake thousands of crimes."
-Sir Syed Ahmad Khan

Lab-III

Course Code: CABSMJ3P03

DEPARTMENT OF COMPUTER SCIENCE
ALIGARH MUSLIM UNIVERSITY, ALIGARH
Welcomes
NAAC Peer Team

DEPARTMENT OF COMPUTER SCIENCE
ALIGARH MUSLIM UNIVERSITY, ALIGARH
कम्प्यूटर साइंस विभाग
شعبۂ کمپیوٹر سائنس
DEPARTMENT OF COMPUTER SCIENCE

B.Sc. (CA) III Semester Lab Manual

DEPARTMENT OF COMPUTER SCIENCE

ALIGARH MUSLIM UNIVERSITY, ALIGARH

2023-2024

Credits

The following lab manual up-gradation committee updated the lab manual:

❒ Prof.AasimZafar(Chairperson)

❒ Dr. Suhel Mustajab

❒ Dr. Arman Rasool Faridi

❒ Dr. Faisal Anwar

❒ Dr. Mohammad Nadeem

The following committee member originally design the lab manual:

❒ Prof. Mohammad Ubaidullah Bokhari

❒ Dr. Arman Rasool Faridi

❒ Dr. Faisal Anwer

❒ Prof. AasimZafar (Converner)

Design & Compilation:

❒ Dr. Shakeel Ahamad

Revised Edition: **June, 2023**

Department of Computer Science,
A.M.U., Aligarh (U.P.) India

COURSE TITLE: Lab -III

COURSE CODE: CABMSJ3P03

CREDIT: 2

PERIODS PER WEEK: 3

CONTINUOUS ASSESSMENT:40

EXAMS: 60

LIST OF CONTENTS (WEEK WISE)

APPENDIX–I

Template for the Index of Lab File

WEEK NO.	PROBLEMS WITH DESCRIPTION		PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
1	1#			
	2#			
	3#			
2	1#			
	2#			
	3#			
3	1#			
	2#			
	3#			

Note: The students should use Header and Footer mentioning their roll no. & name in footer and page no in header.

INTRODUCTION

This assignment on Java is designed for the students of B.Sc. (Computer Applications) -III Semester. Java is one of the most popular object-oriented programming languages. Its applications are enormous and a good understanding of Java will definitely give the students a technical edge. Many open-source editors of Java are available. Some famous of them are Eclipse, NetBeans, BlueJ and JCreator.

OBJECTIVES

After completing this Lab assignment, the students should be able to:

- To develop an understanding of basic concepts of java.
- To get familiarize with Java data types, branching and looping construct, Class, Objects, Inheritance, Abstract Class, Polymorphism, Interface etc.

LAB INSTRUCTIONS

- The students need to submit the solutions of each exercise.
- The students need to ensure that each question is assessed and signed by the Teacher.
- All Students are required to complete the assignment before due date.
- Late submission would not be accepted.
- Cooperate, collaborate and explore for the best individual learning outcomes but **copying is strictly prohibited**.

WEEK #1

Introduction and software requirements to run the java programs:

1# What is the software that helps to run java programs.

2# What is JDK and JRE.

3# What is eclipse IDE.

4# How to run the java program in eclipse/NetBeans IDE.

5# What is the software other than eclipse/NetBeans to run the java programs.

WEEK #2

Basics problems in java

1# Write a java program to add the two numbers.

2# Write a java program to multiply two floating numbers.

3# Write a java program to display a cube of a number.

4# Write a Java program that takes three numbers as input to calculate and print the average of the numbers.

5# Write a Java program to compute the distance between two points.

WEEK #3

Problems Based on if statement/Looping in JAVA

1# Write a java program to check whether the given number is odd or even.

2# Write a java program to find the largest number among the three numbers.

3# Write a Java program that takes a number as input and prints its multiplication table upto 10.

4# Write a Java program to calculate the sum of following series:

1 + 2 + 3 + 4 + ... + N

5# Write a Java program to take a number, divide it by 2 and print the result until the number becomes less than 10.

WEEK #4

Problems Based on Array in JAVA

1# Write a Java program to insert 10, 20, 30 .... in an array and display them.

2# Write a Java program to calculate the sum of all the array elements.

3# Write a java program to print the following pattern.

```
        1
       121
      12321
     1234321
    123454321
```

4# Write a java program to find the sum of following series where n is input by the user.

1+1/2+1/3+1/4+...............+1/n.

5# Write a Java program and compute the sum of the digits of an integer.

6# Write a Java program to calculate the factorial of a number.

WEEK #5

Problems Based on If statement/Looping/Array in JAVA

1# Write a Java program to print the odd numbers from 1 to 99.

2# Write a Java program to check whether a number is prime or not.

3# Write a Java program to swap the first and last elements of an array.

4# Write a Java program to find the maximum and minimum among array elements.

5# Write a Java program to print all prime numbers between 0 to 100

6# Write a Java program to implement linear search.

WEEK #6

Problems Based on If statement/Looping/Array/Strings in JAVA

1# Write a Java program to implement binary search.

2# Write a Java program to arrange the elements of an array in ascending order (Sorting).

3# Write a program to store ‘Java is awesome’ in a string and display it.

4# Write a program to reverse a given string.

5# Write a program to check whether a given string is palindrome or not.

WEEK #7

Problems Based onObject / Class / Constructor in JAVA

1# Create a class FRUIT which has data members color, taste and price. Also create a method display () which will print values of FRUIT object. Create three objects of FRUIT class and call their display () methods.

2# Create a class FRUIT which has data members color, taste and price. It has a method setDetails() which will set the values of color, taste and price. Also create a method display() which will print values of FRUIT object.

3# In previous question, set the values of using color, taste and price using Constructor.

4# Add one-argument constructor and two-argument constructor in addition to default constructor in FRUIT class.

5# Use the concept of constructor-chaining in the previous question using

this ().

WEEK #8

Problems Based on Inheritance in JAVA:

1# Create a class Vehicle, write a method cost() in this class. Create two classes Bus and Train which have their own display() methods and inherit from Vehicle class. Create objects of Bus and Train class and call cost() and display() methods.

2# Create class University which has data members- name and ranking. Create class Faculty that extends University class has data member- name and method- Details(). Create a new class Department which is derived from Faculty and has data member- name, chairman and method- Details() and Display() where Display() method calls Details() methods of both Faculty and Department class in its body. Create an object of Department class to Display() method and University ranking.

WEEK #9

Problems Based on Static methods in JAVA

1# Create class Account (Data members- Id, Account_holder_name, Address; Methodsdeposit(), withdraw()). Create two static methods in Account calculateSimpleInterest() and calculateCompoundInterest() and implement them.

Problems Based on Abstract class in JAVA

1# Create class Account (Data members- Id, Account_holder_name, Address; Methodsdeposit(), withdraw()). Declare deposit() and withdraw() as abstract methods. Declare Account class as abstract. (Create constructor in Account as well).

2# Create two children of Account- Saving (Data Members- Min_balance; Methodsdisplay(), deposit(), withdraw()) and Current (Data Members-Max_withdrawl_limit; Methods-display(),deposit(), withdraw()) . Create constructors for both classes. Implementation of deposit() and withdraw() should be specific to Saving and Current class. Create objects of Saving and Current class and display them.

WEEK #10

Problems Based on Nested Class in JAVA

1# Create class Person (Data Member- name, phone). Create two member inner classes Address (Data Member- House_No, Street, City, State; Method- displayAddr()) and DateOfBirth (Data Member- Day, Month,Year; Method- displayDOB()). Display() is the method of Person class which will display name, address and date of birth of a Person object.

2# Create class Edible. Within that define two static classes Fruit and Vegetable. Fruit class will have two methods- fruitDetails() is a static method and fruitPackaging() is a non-static method. Vegetable class also has similar methods - vegetableDetails() and vegetablePackaging(). Call all the four methods from main method.

Problems Based on Static Polymorphism in JAVA:

1# Create three different minMaxAdd() methods to calculate minimum, maximum and addition of integers, real numbers and characters.

WEEK #11

Problems Based on Dynamic Polymorphism in JAVA:

1# Create a class **ObjectOriented** which has methods- abstraction(), polymorphism() and inheritance(). Create a class **JavaLanguage** which inherits from ObjectOriented class and has its own methods- persistence() and interfaces(). Create an object of **JavaLanguage** class to access all of its own and parent's methods.

2# In previous question, create a new class **C++** which also inherits from **ObjectOriented** class and has its own methods- template() and friendFunction().Create an object of **C++** class to access all of its own and parent's methods.

3# Create class **University** which has data members- name and ranking. Create class **Faculty** that extends **University** class has data member- name and method- Details(). Create a new class **Department** which is derived from **Faculty** and has data member- name, chairman and method- Details() and Display() where Display() method calls Details() methods of both **Faculty** and **Department** class in its body. Create an object of **Department** class to Display() method and University ranking.

WEEK #12

Problems Based on Interface in JAVA:

1# Create an interface **Account** having methods- deposit(), withdraw() and aboutBank() (abountBank() is a static method). Create two classes **Saving** and **Current** which implement the **Account** interface. Call the methods of **Saving** and **Current** classes in main method.

2# In the previous question, create a new method in **Account** interface-takeLoan() (takeLoan() is a default method). takeLoan() method would be implemented by **Saving** class only. Call the methods of **Saving** and **Current** classes in main method.

3# Create interfaces **Bike** and **Scooty**, both of which have two methods-offer() and details() (details() is default method). Create a new class **BuySomething** which implements both interfaces. To remove ambiguity, create a method details() in **BuySomething** class as well in which call the details() method of both interfaces. Call the methods of **BuySomething** class in main method.

WEEK #13

Problems Based on File Handling in JAVA:

1# Write a Java Program to Create a new file.

2# Write a Java Program to Write into a file.

3# Write a Java Program to Copy one file into another file.

4# Write a Java Program to Append a String in an Existing File.

5# Write a java program to find total no. of characters in that file.

6# Write a java program to find total no. of lines in that file.

WEEK #14

Problems Based on Exception Handling and JDBC in JAVA:

1# Write a program that calls a method that throws an exception of type ArithmeticException in a for loop at an undesirable situation (such as divide by zero or taking square root of negative number). Catch the exception and display appropriate message.

2# Create a user-defined exception InvalidAgeException when the age of a person is below 18 years. Use this exception at appropriate place.

3# Write a Java program to display the employee id, age, first name and last name using JDBC connectivity.