Practical 3

1. Write a java program that asks the user to enter number in an array of size ‘n’. then display only the numbers that are divisible by 2 and 3.
2. Make an array of integers of size 30 and store 30 integer in it then display the integers that are between 16 and 47;
3. Define string array of size 10 and insert name of 10 student in it. Then list the name of students that ends with letter ‘a’.
4. Write a program to create integer array of size ‘n’ and insert ‘n’ data on it. Now display the element of array that are in even position
5. Create a String array that holds name of 5 birds and print the name of such birds whose first letter start with ‘s’.
6. Write a program to create multi-dimensional array of size m*n and insert integer value on it. Now display the element of array using for and for each loop
7. Write a program to create two multi-dimensional array of size m*n and insert integer value on it. Perform multiplication operation and store the result in third array and display the content of third array
8. Write a program to demonstrate jagged array

Practical 4

1. Create a class Student having data member roll (int), name (string), address (string), gender (string), faculty (string) and method setData() to set the value of data member and display method that display the data of student. Now create two object of Student and invoke set and display method.
2. Create a class Calculate which contains data member num1 and num2 both in integer and methods setCalc() to set the data, calcSum() that calculate the sum of num1 and num2 and display the result, calcMulti() that calculate the multiplication of num1 and num2 and returns the result, calcDifference that calculate the difference between num1 and num2 and display the result. Now, create some instance of Calculate and invoke all the methods.
3. Create a class Number having instance variable x and y both in integer, default constructor that set the value of x and y to 0, parameterized constructor that sets the value of x and y, method findOdd() that calculates the even no. occurring between x and y and display the result, findEven() that calculates the odd no. occurring between x and y and display the results. Now, create some instance of Number and invoke all the methods.
4. Create a class Rectangle with attributes length and breadth. The class contains method computeArea() and displayArea(). Now, create two object of Rectangle and find area and display the area.
5. Create a class Numbers with three integer instance variables x, y and z. the class will have one constructor to set the value of instance variable and method getMax() that find the greatest number between x, y and z and return the results. Create one object of Number and invoke the method to find greatest number.
6. Create a class Box with attributes length, breadth and height. The class contains method computeVolume() that computes the volume of box and return the volume. Now, create two object of box and find volume and display it.
7. Create a class Product with name, qty and price. Create a parameterized constructor to set the product details. Provide the method getName(), getQty() and getPrice() that return product name, qty and price. Also create method getTotal() that returns the total price. Then create a class ProductDemo with main method that creates two object of Product and find the total price of both products.
8. Create a class Shape that contains instance variable length, breadth and height. Create a default constructor that sets the value of instance variable to zero, constructor with two parameter that will sets the value of length and breadth only and constructor with three parameter that will sets the value of length, breadth and height. After this create calcAreaRectangle() that calculates the area of rectangle, calcVolumeBox() that calculates volume of box and display the result. Now create first object of Shape wihich will have name rectangle and calls constructor with two parameter and calAreaRectangle() method, create second object of Shape that will have name Box which will call constructor with three parameter and calcVolumeBox() method.

Practical 5

1. Create a class Work which consist of four methods with same name calculate (). If two parameters is passed calculate the sum between two parameter and display the result, if three parameter of type double is passed then find the multiplication between three parameter and return the result, if two parameter of float is passed then calculate difference between two parameter and display the result. Now create the instance of Work and invoke all the method to demonstrate method overloading condition.
2. Create a class Student having data member roll, name, address, faculty, grade(string). Create suitable constructor to set the details of student and methods findDivision() that calculates the division of student based on grade achieved and return the division(if grade=A+ then distinction, if grade=A then very good, grade B+ then first division, grade B then second division , grade C then pass in individual subject and if grade D the fail), display() method that display the student details and also display the division achieved by the student in main method. Now create the instance of two students and demonstrate the above scenario.
3. How static polymorphism is achieved in java? Show a program
4. Create an abstract class Calculation which has one normal method int add(int x, int y) which will calculate the sum of x and y and return the results and one abstract method int mul (int x, int y) that performs multiplication. Create a class Demo that inherit abstract class that implement all the abstract method, and it have its own method int calcDiv(int x, int y) that calculates the division between x and y and return the result. Now create the object of Demo and demonstrate the above scenario. After this access the method of abstract class using reference of abstract class.
5. Create an abstract class FMachine having methods getData() and putData(). Derive a class Airplane having instance variable code, name, capacity and methods getData() and putData() that overrides method of abstract class to read and display the result. Create some instances of Airplane and call the required methods.
6. Create an abstract class Calculator having abstract methods calcPrime(int x, int y), calcEven(int x, int y). Now create a class SimpleCalculator that inherit an abstract class and provide the implementation of abstract method

Practical 6 (Inheritance)

1. Provide an example of single inheritance of your choice. (Two child class inheriting single parent class)
2. Provide an example of multi-level inheritance of your choice.
3. Provide example on how constructor of super class is called.
4. Provide example on method overriding condition
5. Provide example on dynamic method dispatch
6. Create a class Book with member variables book_id and pages. Then create a subclass FictionBooks of Book with member variable called name. Create some instance of FictionBook and set the value of both subclass and superclass and display all member variables that have been initialized.
7. Create a class named Shape with a method to print “this is shape”. Then create two other class named “Rectangle” and “circle” inheriting the class Shape both having method to print “this is rectangular” and “this is circular”. Create subclass Square that now inherits Rectangle which have its own method to display “this is square of rectangle”. Now call the method of Shape and Rectangle class by the object of square class inside the class having main method.
8. Create a class ExamDept which hold the details of student like roll, name, address and faculty and also have methods to set the values of student and display the details of student. Create a subclass Result that inherits Exam dept which have data member to hold marks of five subject and method setMarks() to set the marks of 5 subject, calcPercentage() that calculate the total marks, percentage and division achieved by students and display the achieved results. Now create the object of Result and demonstrate the above scenario.
9. Provide an example on method overriding condition (dynamic or run time polymorphism)
10. Create a class EmployeeDetails having data member empId, empName, empGender, empAddress, and empPosition, constructor to set the details and display method to display the details. Create a subclass SalaryInfo that will inherit EmployeeDetails having own data member salary which will record salary per year, constructor to set the value of salary and method calcTax() that deduct the tax on salary and display the final salary. Tax rate is given as (if salary <= 400000 tax is 1%, salary between 400001 to 800000 tax is 10% and salary > 800000 tax 20%). Now create the object of Salary info and demonstrate the scenario.
11. Create a class BankAccount that have data member accNo, Name, address, phone, gender, constructor to set the bank details and showDetails() method to display the account details. Create subclass SavingAccount which inherits BankAccount and have data member currentAmount and month, constructor to set the data member, methods calcTotal() that update the currentAmount by adding the bonus amount and return the result. (if saving month is below 3 mnth bonus is 2%, if saving month is between 3 to 6 month bonus is 5%, if saving month is between 6 to 12 month bonus is 8% and if saving month is above 1 year bonus is 11.03%). Now create the object of SavingAccount and demonstrate the scenario.

```
pritishmashakya@Pritishmas-MacBook-Pro Siddhartha_BIM % javac One.java
pritishmashakya@Pritishmas-MacBook-Pro Siddhartha_BIM % java One
30 + 40: 70
2.4 - 2.6: -0.19999981
Product: 42.432
```

```
pritishmashakya@Pritishmas-MacBook-Pro Siddhartha_BIM % javac Three.java
pritishmashakya@Pritishmas-MacBook-Pro Siddhartha_BIM % java Three
30 + 40: 70
2.5 * 4.5: 11.25
```

```
pritishmashakya@Pritishmas-MacBook-Pro Siddhartha_BIM % javac Five.java
pritishmashakya@Pritishmas-MacBook-Pro Siddhartha_BIM % java Five
Details of airplane 1:
Code: 101
Name: Fighter
Capacity: 100
Details of airplane 2:
Code: 102
Name: Jet
Capacity: 95
```

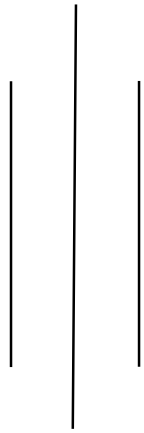
```
pritishmashakya@Pritishmas-MacBook-Pro Siddhartha_BIM %
javac Six.java
pritishmashakya@Pritishmas-MacBook-Pro Siddhartha_BIM %
java Six

Prime numbers from 1 to 50:
2       3       5       7       11      13      17
19      23      29      31      37      41      43      47
Prime numbers from 1 to 10:
2       3       5       7
Even numbers from 1 to 10:
2       4       6       8       10      %
```

```
pritishmashakya@Pritishmas-MacBook-Pro Siddhartha_BIM % javac Two.java
pritishmashakya@Pritishmas-MacBook-Pro Siddhartha_BIM % java Two
Details of student 1:
Name: Siddhartha
Roll No: 23
Address: Dallu
Faculty: BIM
Grade: A+
Division: Distinction
Details of student 2:
Name: Raj
Roll No: 12
Address: Thamel
Faculty: BIM
Grade: D
Division: Fail
```

```
pritishmashakya@Pritishmas-MacBook-Pro Siddhartha_BIM % javac Four.java
pritishmashakya@Pritishmas-MacBook-Pro Siddhartha_BIM % java Four
Sum: 9
Product: 30
Division: 2
```

National College of Computer Studies

Paknajol, Kathmandu



Lab Report On Object Oriented Programming With Java

Assignment No. 6

Submitted by:

Name: Siddhartha Shakya

Program: BIM

Section: B

Roll no: 23

Submitted to:

Mr. Sujesh Manandhar

Mr. Kapil Chhatkuli

Date of Submission: 2024/08/02