OBJECT ORIENTED PROGRAMMING

Practical Assignment

Silver Touch TECHNOLOGIES

Program 1:

Write a class **Marks** with three data members to store three marks. Write three member functions **in()** to input, **sum()** to calculate and return the sum and **avg()** to calculate and return the average marks. Also write the constructor for this class that initialises all data members to zero. In main function, create an object and make use of all functions.

Program 2:

Write a class **Circle** with one data member **radius.** Write three member functions **setRadius()** to set radius value with parameter value, **area()** to calculate and display area of circle and **circum()** to calculate and display circumference of circle. Also write the constructor for this class that initialises all data members to zero. In main function, create an object and make use of all functions.

Program 3:

Write a class **Book** with three data members **BookID**, **Pages** and **Price**. It also contains the following member functions:

- The **get()** function is used to input values
- The **show()** function is used to display values
- The **set()** function is used to set the values of data members using parameters
- The **getPrice() function** is used to return the value of **Price**

The program should create two objects of this class and input values for these objects. The program should display the details of the most costly book.

Program 4:

Write a class **Result** that contains **rollNo**, **name** and **marks** of three subjects. The marks are stored in an array of integers. The class also contains following member functions:

- The **input()** function is used to input values of data members
- The **show()** function is used to display the values of data members
- The **total()** function returns the total marks of a student
- The **avg()** function returns the average marks of a student

The program should create an object of this class and make use of all member functions.

Program 5:

Write a class **Array** that contains an array of integers store five values. It also contains the following member functions:

- The **fill()** function is used to fill the array with values from user
- The **display()** function is used to display the values of array
- The **max()** function shows the maximum value in the array
- The **min()** function shows the minimum value in the array

All the member function should be defined outside class. In main function, create an object of type **Array** and make use of all member functions.

Program 6:

Create a class called employee that contains two members: an employee number (type int) and the employee's compensation (in rupees : type float). Member functions should allow the user to enter this data and display it.

Program 7:

Define a class for Bank Account that includes following data members:

- Name of account holder
- Account number
- Type of Account (i.e. Current, Saving, etc.)
- Balance amount in the account

The class also contains the following member functions:

- A constructor to assign the initial values
- **createAccount()** function to assign all data members through parameters
- **deposit()** function to deposit some amount. It should accept the amount as parameter
- **withdraw()** function to withdraw an amount after checking the balance. It should accept the amount in parameter
- **display()** function to display all account details

In main function, create an object and make use of all functions. The program should display a proper menu to the user and perform the appropriate function according to the user choice. The menu should be continuously displayed to the user again and again user gives proper choice to exit the program.

Program 8:

Write a class **Run** that contains the following data members:

- The name of the runner
- The Distance covered by the runner

This class has following member functions:

- Get function to input runner name and distance
- Show function to display runner name and distance

The user should be able to show the name of the runner who has covered the longest distance at any point of time. Use appropriate function in class to implement this functionality.

Program 9:

Write a class **Car** that contains the following attributes:

- The name of car
- The direction of car (i.e. E, W, N, S)
- The position of car (from imaginary zero point)

The class has following member functions:

- A constructor to initialise the attributes
- Turn function to change the direction of the car to one step right side(e.g. if the direction is to E , it should be changed to S and so on)
- Overload the turn function to change the direction to any side directly. It should accept the direction as parameter.
- Move function to change the position of car away from zero point. It should accept the distance as parameter.
- Show function to display all data for a car

In main function, create an object and make use of all functions. The program should display a proper menu to the user and perform the appropriate function according to the user choice. The menu should be continuously displayed to the user again and again until user gives proper choice to exit the program.

Program 10:

Imagine a tollbooth at a bridge. Cars passing by the booth are expected to pay 50 rupees toll. Mostly they do, but sometimes a car goes by without paying. The tollbooth keeps track of the number of cars they have gone by, and of total amount of money collected.

Model this tollbooth with a class called TollBooth. The two data Items are a type unsigned int to hold the total number of cars and a type int to hold the total amount of money collected. A constructor initialises both of these to zero. A member function called payingCar() increment the car total and adds 50 to the cash total. Another function called noPayCar() Increments 1he car total but adds nothing to the cash total. Finally, a member function called display() displays the two totals. Make appropriate member function const.

In main function, create an obJect of this class. The program should display a proper menu to the user and perform the appropriate function according to the user choice. The menu should be continuously displayed to the user again and again until user gives proper choice to exit the program.

Program 11:

Create a class called Time that has separate int data members for hours, minutes and seconds. One constructor should initialise this data to 0 and another should initialise it to parameterised values. A member function should display It. In 11:59:S9 format. Another member function should adjust the time values If they exceed by their limits {e.g. adjust seconds if they are greater than 59 and add relative seconds to minutes}. The final member function should add two objects of type time passed as arguments.

A **main()** program should create two initialised **time** objects and.one that isn't initialised. Then It should add the two initialised values together, leaving the result In third **time** variable. Adjust the time values of this third variable and finally display its values. Make appropriate member function **const**.

Program 12:

Create a class that includes a data member that records a count of how many objects have been created so far. {This member should be applied to the class as a whole; not to individual objects, What keyword specifies this?) Then, as each object is created, its constructor can examine this count member variable to determine the appropriate serial number for the new object. Add a member function that permits an object to report Its own serial number.

Write **main()** program that creates three objects and queries each one about its serial number. They should respond **I am object number 2**, and so on.

Program 13:

Write a class **Student** that contains the following data members:

- Static variables to create unique members
- A variable to store roll number
- A variable to store name of the student
- An array to store marks of 5 subjects

The class should have following member functions:

- A constructor with no parameters that increments static variable and assign it to roll number. Remaining data members should be initialised to zero or empty values.
- **input()** function to input name and marks of 5 subjects from user
- **show()** function to display roll number, name and marks of 5 subject to the user
- **totalStudents()** function that should display the number of objects that has been created so far. This function should be static.
- **totalMarks()** function that should calculate and return total marks of a student.
- **getHighest()** function that returns highest marks of the student
- **getLowest()** function that returns lowest marks of the student
- **getAverage()** function that returns average marks of student
- **getPassCount()** function that counts and returns total number of subjects in which the student is passed. (A student is passed if he/she gets 50% or above marks)

In main function, create three objects, input their values and display them. For each student, display his/her total marks, highest, lowest, average marks and the number of subjects in which he/she is passed. Also display how many students have been registered.

Program 14:

Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' without any parameter in its constructor.

Program 15:

Write a Print the average of three numbers entered by user by creating a class named ‘Average' having a method to calculate and print the average.

Program 16:

Print the sum, difference and product of two complex numbers by creating a class named 'Complex' with separate methods for each operation whose real and imaginary parts are entered by user.

Program 17:

Write a program by creating an 'Employee' class having the following methods and print the final salary.

- **getInfo()** which takes the salary, number of hours of work per day of employee as parameter
- **addSalary()** which adds ₹10 to salary of the employee if it is less than ₹500.
- **addWork()** which adds ₹5 to salary of employee if the number of hours of work per day is more than 6 hours.

Program 18:

Write Create a class called 'Matrix' containing constructor that initialises the number of rows and number of columns of a new Matrix object. The Matrix class has the following information:

- number of rows of matrix
- number of columns of matrix
- elements of matrix in the form of 2D array

The Matrix class has methods for each of the following:

- Get the number of rows
- Get the number of columns
- Set the elements of the matrix at given position (i,j)
- Adding two matrices. If the matrices are not addable, "Matrices cannot be added" will be displayed
- Multiplying the two matrices

Program 19:

Write a program that has variables to store Car data like; CarModel, CarName, CarPrice and CarOwner. The program should include functions to assign user defined values to the above mentioned variable and a display function to show the values. Write a main that calls these functions. Now write another runner class that declares three Car objects and displays the data of all three.

Program 20:

A Student is an object in a university management System. Analyse the concept and identify the data members that a Student class should have. Also analyse the behaviour of student in a university management System and identify the methods that should be included in Student class.

Program 21:

Create a class circle class with radius as data member. Create two constructors (no argument, and two arguments) and a method to calculate Circumference.

Program 22:

Create a class Account class with balance as data member. Create two constructors (no argument, and two arguments) and methods to withdraw and deposit balance.

Program 23:

Create Write a class Time with three data members to store hr, min and seconds. Create two constructors and apply checks to set valid time. Also create display function which displays all data members.

Program 24:

Create Implement Vehicle as outer and owner as the inner class, the vehicle class contains vehicle name, engine cc, model as data members. The inner class data members are owners name, CNIC number and phone contact of the owner. Write down proper setters/ getters and constructors for both the classes.

- Override the method of a class using anonymous inner class.
- Pass an anonymous inner class as a method argument.
- Implement the inner class as static first and then as non static nested class.

Program 25:

Create an Abstract class Student that contains a method take exam, implement the method in the child classes PhdStudent and GradStudent in which PhdStudent takes exam by giving his final defence presentation while the graduate student gives a written paper

Program 26:

Define a class called Fraction. This class is used to represent a ratio of two integers. Create two constructors, set, get and display function. Include an additional method, equals, that takes as input another Fraction and returns true if the two fractions are identical and false if they are not.

Program 27:

Create abstract class called Language which has one method as getLanguage() with empty method body which has String parameter as name. Create Child class which overrides method getLanguage(). In main method access getLanguage() method to display language name.

Program 28:

Write a program to create interface called Polygon which has one method with empty body as getArea() and one default method as getSides(). Create classes of Rectangle and Square to inherit the properties & methods of Polygon class. Override the methods of Polygon class in child classes. In main method access both methods to display area and sides from the child classes.

Program 29:

Write a program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.

Program 30:

Create interface called Engine with two methods as changeGear() which accepts int parameter and has no body also speedUp() which accepts int parameter and also has no body. Create Vehicle class which implement and override methods of the Engine interface. In main method create instance of the Vehicle class to access both methods and print the values of speed as well as gear.