|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | Write a program that uses a for loop to find the sum of all even numbers from 1 to 100. | For loop. If (number % 2 == 0) then it is even |
|  | Write a program that uses a do-while loop to display numbers from 1 to 10. | Do-while |
|  | Write a program that prints the numbers from 1 to 10 in reverse order.(simple) | While loop |
|  | Write a program that uses a while loop to print the Fibonacci series up to 20(Complex example) | While loop. Fibnocci series is starts with 0 and 1, and the subsequent numbers are generated by adding the two previous numbers  0 1 2 3 5 8 13 |
|  | Write a program that uses a switch statement to display the day of the week based on a user input number. | switch |
|  | Create a class named "Calculator" with methods for addition, subtraction, multiplication, and division. Implement method overloading to handle different parameter types (int, double) for each operation.  i.e override method addition and substract methods to take 3 or 4 parameters | Method overloading(static polymorphism/static method binding) |
|  | Create a subclass named "ScientificCalculator" that extends the Calculator class. Override the multiplication method to calculate the power of two numbers. | Inheritance and Method overloading |
|  | Checking if a string is a palindrome(complex) | String Operations. A palindrome is a word, phrase, number, or sequence of characters that reads the same forward and backward |
|  | Write a program that accepts a sentence as input from the user and counts the number of words in the sentence. Display the total count at the end. | String operations. Split the string by spaces |
|  | Create a class named "Person" with the following attributes: name, age, and email. Implement a method to display the details of the person. | Class, instance variables, and methods |
|  | Create a subclass named "Employee" that inherits from the Person class. Add an additional attribute called "employeeId" to the Employee class. Implement a method to display the employee details. | Inheritance, superclass, and subclass |
|  | Determining if an employee is eligible for retirement if you send Employee. Put the method name as “isEmployeeEligibleForRetirement” | boolean |
|  |  |  |
|  | Create a class named "Organization" with an ArrayList of type Person to store the employees. Implement a method to add employees to the organization. |  |
|  | Create an abstract class named "Vehicle" with the following attributes and methods:  Attributes:  brand (String)  price (double)  Methods:  A constructor to initialize the brand and price.  An abstract method named "displayInfo()" that displays the information about the vehicle.  Create two subclasses named "Car" and "Motorcycle" that inherit from the Vehicle class. Implement the displayInfo() method in each subclass to display specific information about the vehicle type | AbstractClass |
|  | Create an interface named "Person" with a method called "displayInfo".  Implement the interface in the Customer and Employee class.  Customer class alone should have makePayment method should display a message indicating that the payment has been made.  Invoke displayInfo by create instance of Customer and Employee and reference to person  MalePayment should be invoked only by Customer | Interface |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |