

classes and Objects

1. . Create a class called Saledetails which has data members like Salesno, Productno, Price, dateofsale, Qty, TotalAmount

Create a method called Sales() that takes qty, Price details of the object and updates the TotalAmount as $Qty * Price$

Pass the other information like SalesNo, Productno, Price,Qty and Dateof sale through constructor

call the show data method to display the values.

Hint : Use This pointer

2. Create a class called Customer with Customerid, Name, Age, Phone,City.

Write a constructors with no arguments and another constructor with all information. Write a method called DisplayCustomer(), which is called directly without any object. Also include destructor

General

3. Create a Class Program which would be used to accepts two Strings - FirstName and LastName and call the static method Display() that displays the first name in one line and the LastName in the second line after converting the same to upper case.

4. Create a Program to count the no. of occurrences of a letter in a given string (for example in a string called "OOPS PROGRAMMING", O appears 3 times)

Hint : Accept a string and also the letter to be counted

5. Create a Console Program to accept numbers and print the sum of the digits.

6. Create a class called Scholarship which has a function Public void Merit() that takes marks and fees as an input.

If the given mark is ≥ 70 and ≤ 80 , then calculate scholarship amount as 20% of the fees

If the given mark is > 80 and ≤ 90 , then calculate scholarship amount as 30% of the fees

If the given mark is > 90 , then calculate scholarship amount as 50% of the fees.

In all the cases return the Scholarship amount

Properties

7. Create a Class called Doctor with RegnNo, Name, Feescharged as Private members. Create methods to give values and also to display the same.

Indexers

8. Create a class called Books with BookName and AuthorName members. Instantiate the class through constructor and also write a method Display() to display the details. Create an Indexer of Books Object to store 5 books in a class called BookShelf. Using the indexer method assign values to the books and display the same.

Structures

9. Create a structure

struct Books

private string title;

private string author;

private string subject;

```
private int book_id;
```

Include 2 methods called GetValues() and ShowValues() to accept and display details.

create a Class called TestStructures that invokes the above methods of the structure to accept and display the details of the book

Overloading

10. Create a Program class that has method to swap 2 numbers or 2 strings

11. Write a class Box that has Length and breadth as its members. Write a function that adds 2 box objects and stores in the 3rd. Create a Test class to execute the above.

Inheritance / Overriding

12. Create an Abstract class Student with Name, StudentId, Grade as members and also an abstract method Boolean Ispassed(grade) which takes grade as an input and checks whether student passed the course or not.

Create 2 Sub classes Undergraduate and Graduate that inherits all members of the student and overrides Ispassed() method

For the UnderGrad class, if the grade is above 70.0, then isPassed returns true, otherwise it returns false. For the Grad class, if the grade is above 80.0, then isPassed returns true, otherwise returns false.

13. Create an Employee class with Empid int, Empname string, Salary float. Pass values to the members through Constructor.

Create a derived class called ParttimeEmployee with Wages as a data member. Instantiate the base class through the derived class constructor

Exception Handling

14. Create a console application to book train tickets. Create a Passenger class with (Name, Age) and write a function called TicketBooking(no_of_tickets) that takes no.of tickets to be booked. If the no of tickets is > 2 per booking, raise an user defined exception, and print "cannot book more than 2 tickets". Else Print "Ticket Booked Successfully". Add a Test class to call TicketBooking method by accepting all required details.

15. Write a class called LoanProcess with Loan_No, Customer Name, LoanAmount, EMI_Amount, Account_Balance as its members. Create a method calculate_EMI() for the LoanAmount , with the rate of interest as 13% for a total of 3 years and store it in the EMI_Amount. The rest of the information to be passed through constructors. Write another function CheckBalance() which checks if the Account_Balance is less than the EMI_AMount. If yes then throw a custom exception. Display " Not Sufficient Balance to repay Loan" in the finally. Give explanatory comments.

Collections

16. Write a program to find the Sum and the Average points scored by the teams in the IPL. Create a Class called Cricket that has a function called Pointscalculatation(int no_of_matches) that takes no.of matches as input and accepts that many scores from the user. The function should then display the Average and Sum of the scores

17. Create a Class called Products with Productid, Product Name, Price. Accept 10 Products, sort them based on the price, and display the sorted Products

18 Create a simple Stationery application to add items and display added items using Generic collections

Interfaces

19. Create an Interface IStudent with StudentId and Name as Properties, ShowDetails() as its method. Create 2 classes Dayscholar and Resident that implements the interface Properties and Methods.

20. Create a class library CalculateConcession () that takes age as an input and calculates concession for travel as below:

If age<=5 then “Little Champs- Free Ticket” should be displayed

If age >60 then calculate 30% concession on the totlfare(Which is a constant Eg:500/-) and Display “ Senior Citizen” + Calculated Fare

Else “Print Ticket Booked” + Fare.

Create a Console application with a Class called Program which has TotalFare as Constant, Name, Age. Accept Name, Age from the user and call the CalculateConcession() function

ADO Programming

1. Create a Class to connect to the Employeedatabase . Add a method DisplayEmployee () which will select all records from the employee table and display the same.
2. Accept Dept Id and Deptname values from the user as parameters and add a record to the Department table after establishing a connection with the databse. After adding display “Successfully added one record”.

3. Write steps with required classes and objects to establish a connection to the Employee Database
4. Create a program to accept Employee Id from the user. Search the Value in the table Employee and if found display a message “ Are you sure to delete this record Y/N”. If Y then delete the record
5. Write a console application that connect to the Employee Database and call a procedure (PrcDisplaySalary) in the table by passing an Employee id.

Hint: Create the procedure PrcDisplaySalary which accepts Empid and displays his/her salary