

Q1.to ecommerce web site provides cashback on the purchase amount

Below Rs-1000 no cashback

1000

50005% cash back

5001-10000

10% cashback

16V 1000 15% cashback take the input amt. from the user and calculate the cashback amt.

```
using System;
class Program{
    static void Main(){
        double PA, cashback = 0.0;
        Console.Write("Enter the purchase amount (in Rs.): ");
        PA = Convert.ToDouble(Console.ReadLine());
        if (PA<1000){
            cashback = 0.0; }
        else if (PA >= 1000 && PA < 5000){
            cashback = 0.05 * PA;
        }
        else if (PA >= 5000 && PA < 10000){
            cashback = 0.10 *PA;
        }
        else{
            cashback = 0.15 * PA;
        }
        Console.WriteLine($"Your Cashback is : Rs. {cashback:F2}");
    }
}
```

Q2. A company gve bonus to employee based on their year of service less than 5 year no bonus

5-10- 10% of salary

10-20 20% of Salary

Above 20 yrs -30% of salary

Take the input of salary and yrs of service from user side

```
using System;
class Program{
```

```

static void Main(){
    Console.WriteLine("Enter the num of years of service: ");
    int yos = int.Parse(Console.ReadLine());
    Console.WriteLine("Enter the salary: ");
    double salary = double.Parse(Console.ReadLine());
    double bonus = 0;
    if(yos >= 5 && yos<= 10){
        bonus = salary * 0.10;
    }
    else if(yos >10 && yos <= 20){
        bonus = salary * 0.30;
    }
    else if (yos > 20){
        bonus = salary * 0.30;
    }
    Console.WriteLine("The Bonus is: "+ bonus);
}
}

```

Q3.In a college corpus a structured parking syphen is required to based on efficiently manage parking fees of people.. collection

different Category The parking system must classify vehicle in 2-5years

- 1) VIP I faulty
- No parking fees is charged
- 2) Student Car Car-
10 Rs. parking fees will be charged
- 3)for visitor car
20 Rs will be charged

The system should allows user to register a vehicle enter automatically collect the collection and generate a summary report the the no. vehicles parked in each Category and total revenue

```

using System;
using System.Collections.Generic;

public class Vehicle
{
    public string VehicleNumber { get; set; }
    public string VehicleType { get; set; }
    public int ParkingFee { get; set; }
}

```

```

public class ParkingSystem
{
    public List<Vehicle> Vehicles { get; set; }
    public decimal TotalRevenue { get; set; }

    public ParkingSystem()
    {
        Vehicles = new List<Vehicle>();
    }

    public void RegisterVehicle()
    {
        Console.Write("Enter vehicle number: ");
        string vehicleNumber = Console.ReadLine();
        Console.Write("Enter vehicle type (VIP, Faculty, Student, Visitor): ");
        string vehicleType = Console.ReadLine();
        Vehicle vehicle = new Vehicle { VehicleNumber = vehicleNumber, VehicleType =
vehicleType };
        switch (vehicleType)
        {
            case "VIP":
            case "Faculty":
                vehicle.ParkingFee = 0;
                break;
            case "Student":
                vehicle.ParkingFee = 10;
                break;
            case "Visitor":
                vehicle.ParkingFee = 20;
                break;
        }
        Vehicles.Add(vehicle);
        Console.WriteLine("Vehicle registered successfully!");
    }

    public void CollectParkingFee()
    {
        foreach (var vehicle in Vehicles)
        {
            TotalRevenue += vehicle.ParkingFee;
            Console.WriteLine($"Parking fee for {vehicle.VehicleNumber} is
{vehicle.ParkingFee}");
        }
    }

    public void GenerateSummaryReport()
    {
        int vipCount = 0;
    }
}

```

```

int facultyCount = 0;
int studentCount = 0;
int visitorCount = 0;
foreach (var vehicle in Vehicles)
{
    switch (vehicle.VehicleType)
    {
        case "VIP":
            vipCount++;
            break;
        case "Faculty":
            facultyCount++;
            break;
        case "Student":
            studentCount++;
            break;
        case "Visitor":
            visitorCount++;
            break;
    }
}
Console.WriteLine("Summary Report:");
Console.WriteLine($"VIP Vehicles: {vipCount}");
Console.WriteLine($"Faculty Vehicles: {facultyCount}");
Console.WriteLine($"Student Vehicles: {studentCount}");
Console.WriteLine($"Visitor Vehicles: {visitorCount}");
Console.WriteLine($"Total Revenue: {TotalRevenue}");
}
}

```

```

class Program
{
    static void Main()
    {
        ParkingSystem parkingSystem = new ParkingSystem();
        while (true)
        {
            Console.WriteLine("Parking System Menu:");
            Console.WriteLine("1. Register Vehicle");
            Console.WriteLine("2. Collect Parking Fee");
            Console.WriteLine("3. Generate Summary Report");
            Console.WriteLine("4. Exit");
            Console.Write("Enter your choice: ");
            string choice = Console.ReadLine();
            switch (choice)
            {
                case "1":
                    parkingSystem.RegisterVehicle();

```

```
        break;
    case "2":
        parkingSystem.CollectParkingFee();
        break;
    case "3":
        parkingSystem.GenerateSummaryReport();
        break;
    case "4":
        return;
    default:
        Console.WriteLine("Invalid choice. Please try again.");
        break;
    }
}
}
}
```