Introduction

This assignment demonstrates the application of object-oriented programming concepts using C#. It focuses on designing classes with private data members, constructors, and member functions. Task-3 implements a student grade evaluation system based on marks and attendance, while Task-4 develops a vehicle parking fee management system based on vehicle type and parking duration.

Objective

- To understand and apply object-oriented programming concepts in C#
- To create classes using private data members and constructors
- To implement decision-making logic using conditional statements
- To display processed information such as grades and parking fees

Task 3: Student Grade Evaluation System

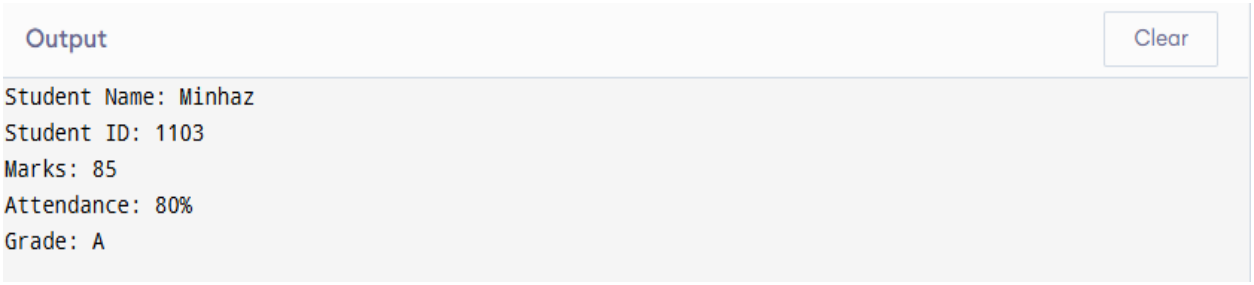
Source Code

```
using System;

class Student

{

   // Private data members

   private string studentName;

   private int studentID;

   private int marks;

   private float attendancePercentage;

   // Constructor

   public Student(string name, int id, int m, float att)

   {

      studentName = name;

      studentID = id;
```

```
        marks = m;
        attendancePercentage = att;
    }
   // Function to calculate grade
    public char GetGrade()
    {
        if (marks >= 80 && attendancePercentage >= 75)
            return 'A';
        else if (marks >= 80 && attendancePercentage < 75)
            return 'B';
        else if (marks >= 60 && marks <= 79 && attendancePercentage >= 75)
            return 'B';
        else if (marks >= 60 && marks <= 79 && attendancePercentage < 75)
            return 'C';
        else
            return 'F';
    }
   // Display function
    public void Display()
    {
        Console.WriteLine("Student Name: " + studentName);
        Console.WriteLine("Student ID: " + studentID);
        Console.WriteLine("Marks: " + marks);
        Console.WriteLine("Attendance: " + attendancePercentage + "%");
        Console.WriteLine("Grade: " + GetGrade());
    }
}
class Program
{
```

```
    static void Main(string[] args)
    {
        Student s1 = new Student("Minhaz", 1103, 85, 80);
        s1.Display();


        Console.ReadLine();
    }
}
```

Output:



Task 4: Vehicle Parking Fee Management System

Source Code

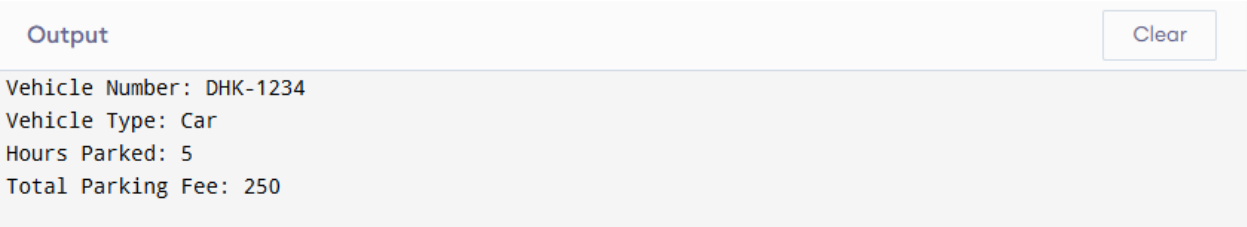
```
using System;


class Vehicle
{
    private string vehicleNumber;
    private string vehicleType;
    private int hoursParked;
    private float parkingFee;


    public Vehicle(string number, string type, int hours)
    {
```

```
    vehicleNumber = number;
    vehicleType = type;
    hoursParked = hours;
    CalculateFee();
}

public string GetVehicleNumber() { return vehicleNumber; }
public void SetVehicleNumber(string number) { vehicleNumber = number; }

public string GetVehicleType() { return vehicleType; }
public void SetVehicleType(string type) { vehicleType = type; }

public int GetHoursParked() { return hoursParked; }
public void SetHoursParked(int hours)
{
    hoursParked = hours;
    CalculateFee();
}

public float GetParkingFee() { return parkingFee; }

private void CalculateFee()
{
    if (vehicleType == "Bike")
    {
        if (hoursParked <= 2)
            parkingFee = 50;
        else
            parkingFee = 50 + (hoursParked - 2) * 20;
```

```
        }
        else if (vehicleType == "Car")
        {
            if (hoursParked <= 2)
                parkingFee = 100;
            else
                parkingFee = 100 + (hoursParked - 2) * 50;
        }
        else if (vehicleType == "Bus")
        {
            parkingFee = 300 + hoursParked * 100;
        }
    }

    public void DisplayBill()
    {
        Console.WriteLine("Vehicle Number: " + vehicleNumber);
        Console.WriteLine("Vehicle Type: " + vehicleType);
        Console.WriteLine("Hours Parked: " + hoursParked);
        Console.WriteLine("Total Parking Fee: " + parkingFee);
    }
}

class Program
{
    static void Main()
    {
        Vehicle v = new Vehicle("DHK-1234", "Car", 5);
        v.DisplayBill();
```

```
    }
}
```

Output:



Conclusion

The vehicle parking system correctly calculates parking fees based on vehicle type and parking duration. The program demonstrates effective use of constructors, conditional logic, and object-oriented principles in C#.