

Taazaa Training

“Assignment - 4”

Topics:-

- Day4A (Professional way)
- enum
- Interface (3 Programs)
- Abstract class (3 programs)
- Constructor chaining

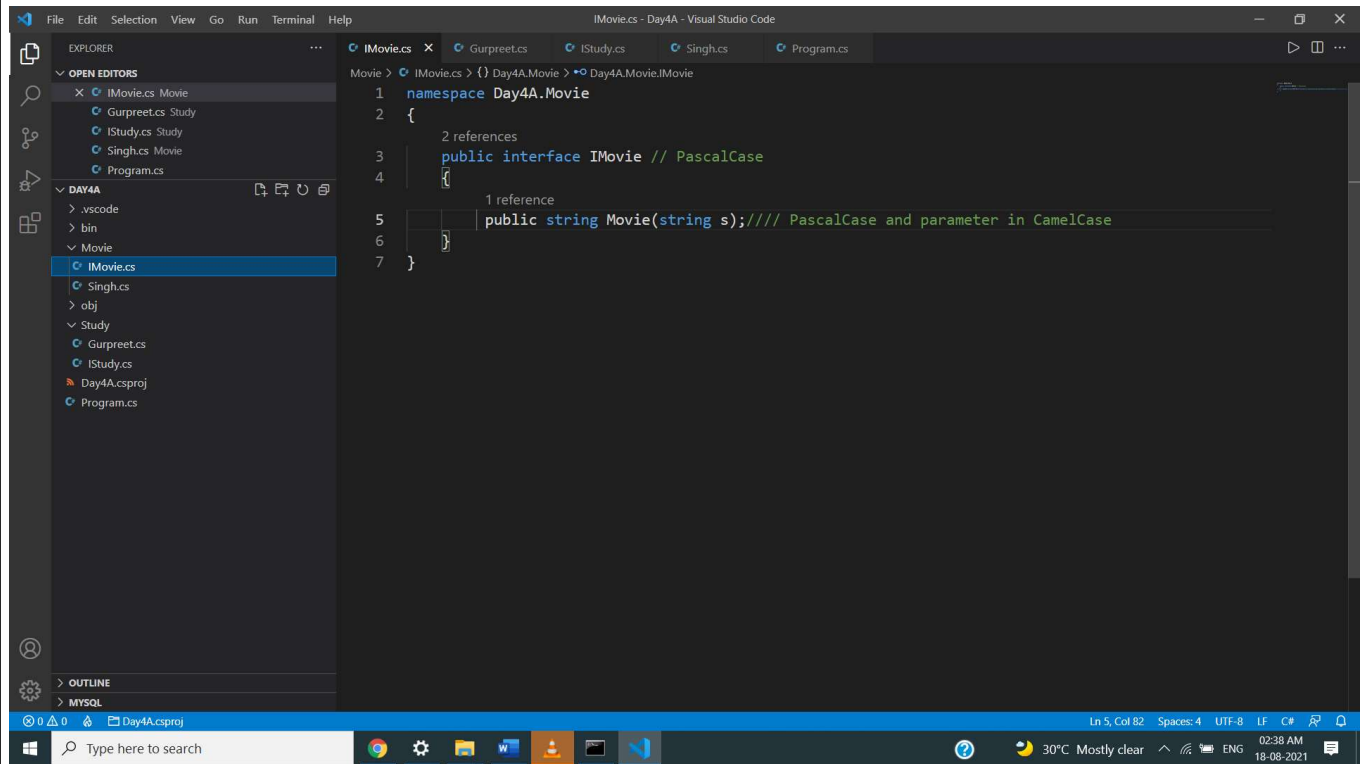
Submitted by: -

Gurpreet Singh

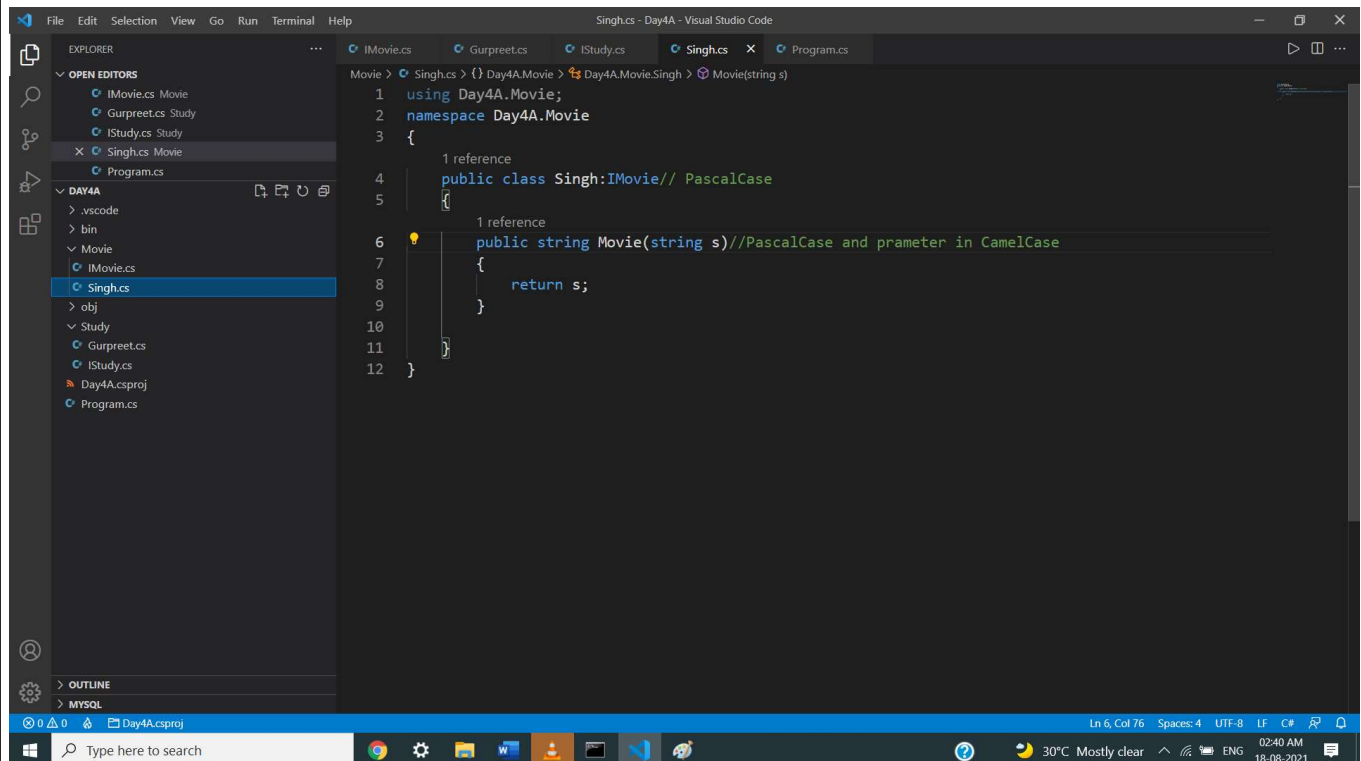
Day4A(Professional way).

Inside Movie folder:-

Interface:-

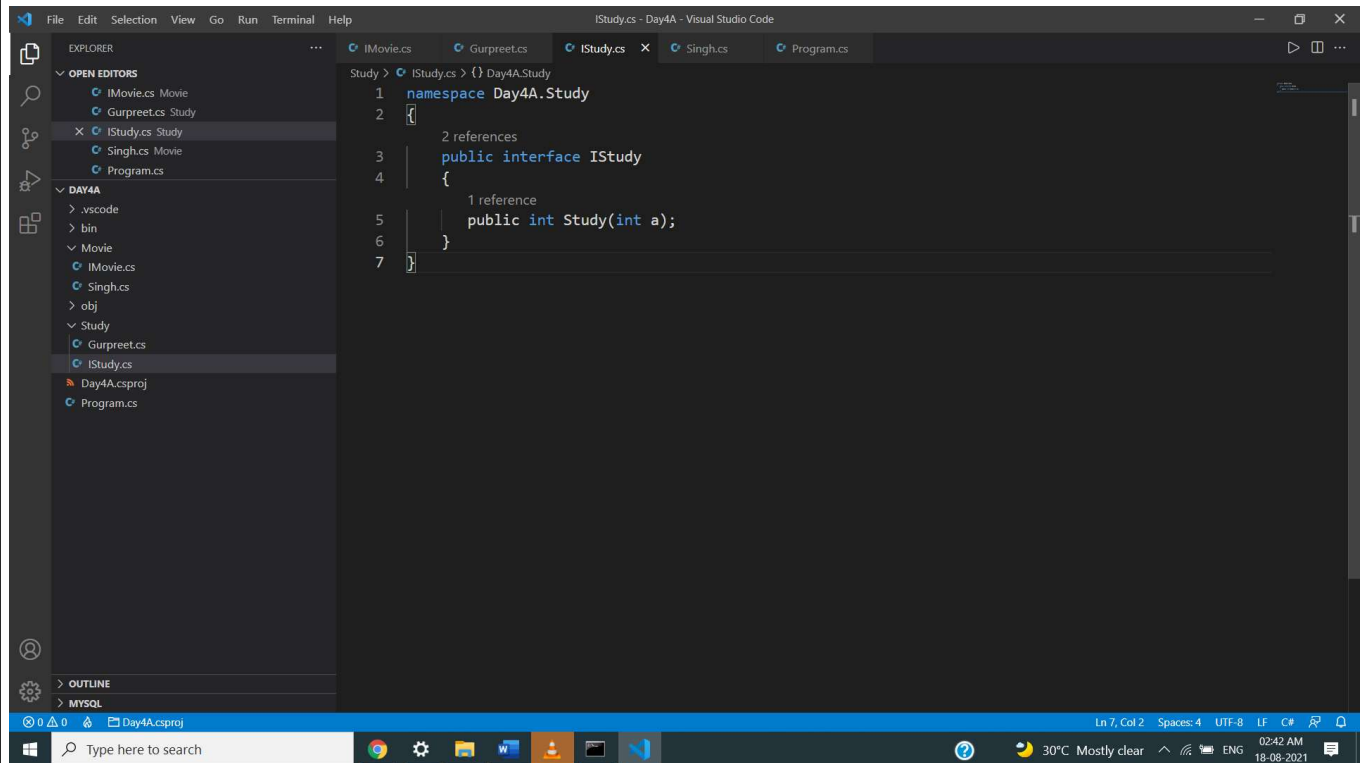


Class:-



Inside Study folder:-

Interface:-

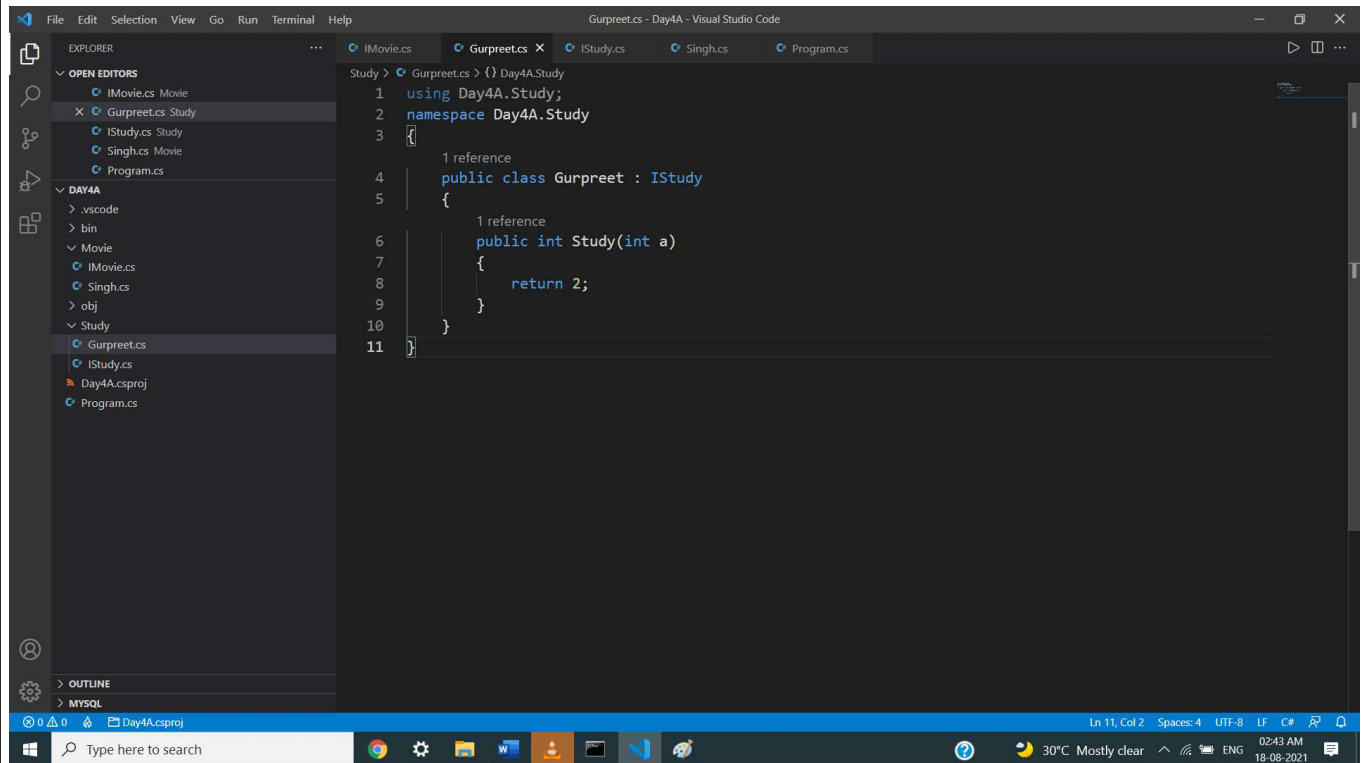


The screenshot shows the Visual Studio Code interface with the 'IStudy.cs' file open. The Explorer pane on the left shows the project structure for 'Day4A', including folders like '.vscode', 'bin', 'Movie', and 'Study', and files like 'IMovie.cs', 'Singh.cs', 'Gurpreet.cs', 'IStudy.cs', and 'Program.cs'. The main editor displays the following C# code:

```
1 namespace Day4A.Study
2 {
3     2 references
4     public interface IStudy
5     {
6         1 reference
7         public int Study(int a);
8     }
9 }
```

The status bar at the bottom indicates the file is at Line 7, Column 2, with 4 spaces, UTF-8 encoding, and LF line endings. The system tray shows the time as 02:42 AM on 18-08-2021.

Class:-



The screenshot shows the Visual Studio Code interface with the 'Gurpreet.cs' file open. The Explorer pane on the left shows the project structure for 'Day4A'. The main editor displays the following C# code:

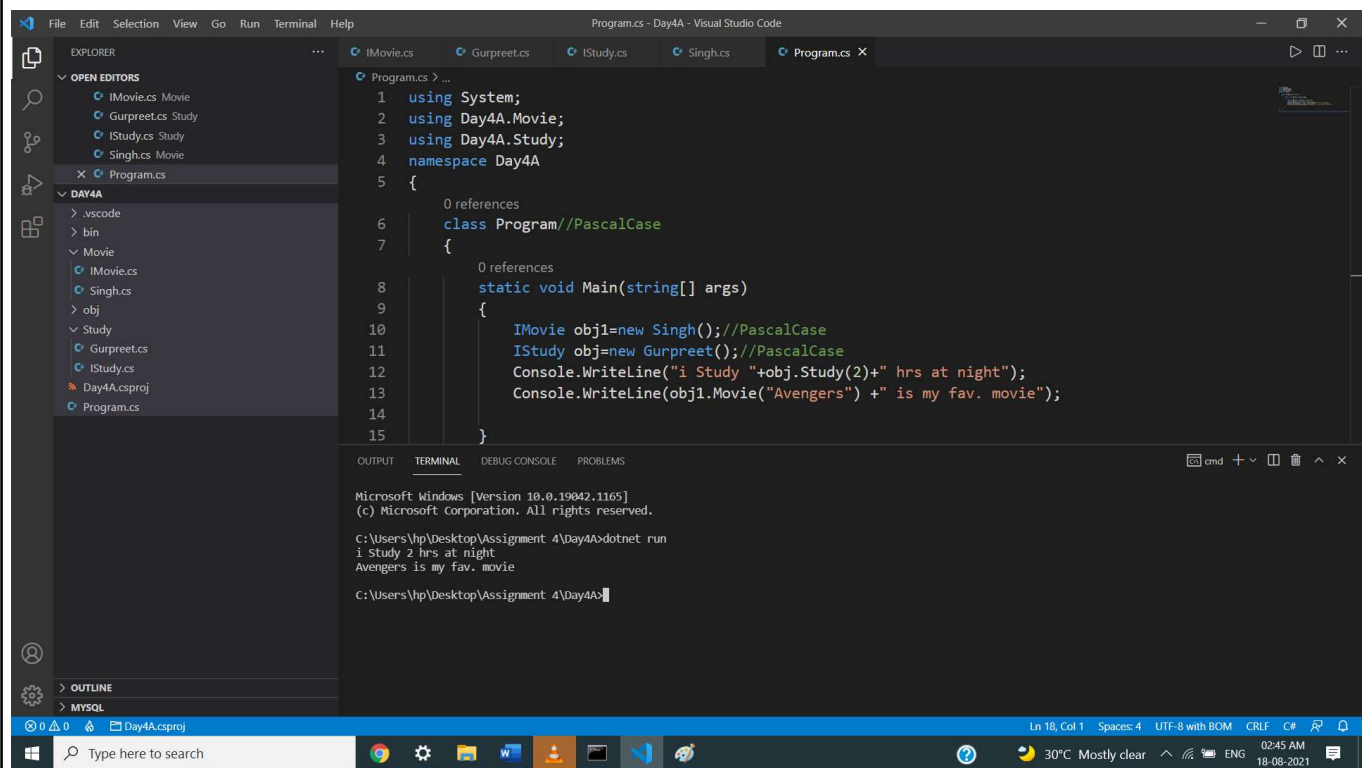
```
1 using Day4A.Study;
2 namespace Day4A.Study
3 {
4     1 reference
5     public class Gurpreet : IStudy
6     {
7         1 reference
8         public int Study(int a)
9         {
10             return 2;
11         }
12 }
```

The status bar at the bottom indicates the file is at Line 11, Column 2, with 4 spaces, UTF-8 encoding, and LF line endings. The system tray shows the time as 02:43 AM on 18-08-2021.

Source Code: -

```
using System;
using Day4A.Movie;
using Day4A.Study;
namespace Day4A
{
    class Program//PascalCase
    {
        static void Main(string[] args)
        {
            IMovie obj1=new Singh();//PascalCase
            IStudy obj=new Gurpreet();//PascalCase
            Console.WriteLine("i Study "+obj.Study(2)+" hrs at night");
            Console.WriteLine(obj1.Movie("Avengers") +" is my fav. movie
");
        }
    }
}
```

Screen Shorts:-



The screenshot displays the Visual Studio Code interface with a C# program open. The Explorer sidebar on the left shows the project structure, including files like IMovie.cs, Gurpreet.cs, IStudy.cs, Singh.cs, and Program.cs. The main editor window shows the code for Program.cs, which includes using statements for System, Day4A.Movie, and Day4A.Study, and a namespace declaration for Day4A. The code defines a class Program with a Main method that creates instances of IMovie, IStudy, and Gurpreet, and prints their details to the console.

```
1 using System;
2 using Day4A.Movie;
3 using Day4A.Study;
4 namespace Day4A
5 {
6     0 references
7     class Program//PascalCase
8     {
9         0 references
10        static void Main(string[] args)
11        {
12            IMovie obj1=new Singh();//PascalCase
13            IStudy obj=new Gurpreet();//PascalCase
14            Console.WriteLine("i Study "+obj.Study(2)+" hrs at night");
15            Console.WriteLine(obj1.Movie("Avengers") +" is my fav. movie");
16        }
17    }
```

The OUTPUT window at the bottom shows the execution results:

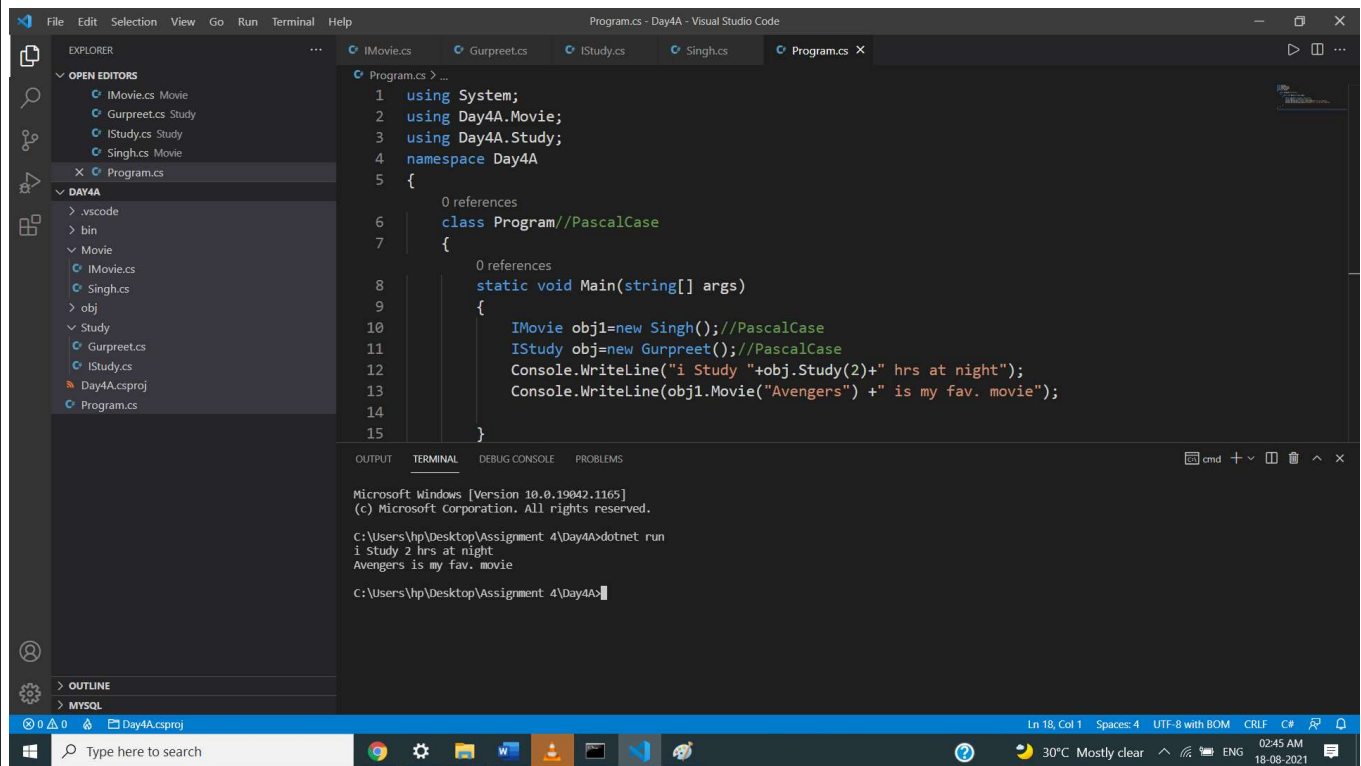
```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\Desktop\Assignment 4\Day4A>dotnet run
i Study 2 hrs at night
Avengers is my fav. movie

C:\Users\hp\Desktop\Assignment 4\Day4A>
```

The status bar at the bottom indicates the file is Program.cs, line 18, column 1, with 4 spaces, using UTF-8 encoding with BOM, CRLF line endings, and the C# language.

Output :-



The screenshot displays the Visual Studio Code interface with a C# program open in the editor. The Explorer sidebar on the left shows the project structure, including files like IMovie.cs, Gurpreet.cs, IStudy.cs, Singh.cs, and Program.cs. The main editor window shows the code for Program.cs, which includes using statements for System, Day4A.Movie, and Day4A.Study, and a namespace declaration for Day4A. The code defines a class Program with a static Main method that creates instances of IMovie, IStudy, and Gurpreet, and prints their details to the console. The Output window at the bottom shows the execution results, including the command prompt path and the output of the program.

```
1 using System;
2 using Day4A.Movie;
3 using Day4A.Study;
4 namespace Day4A
5 {
6     0 references
7     class Program//PascalCase
8     {
9         0 references
10        static void Main(string[] args)
11        {
12            IMovie obj1=new Singh();//PascalCase
13            IStudy obj=new Gurpreet();//PascalCase
14            Console.WriteLine("i Study "+obj.Study(2)+" hrs at night");
15            Console.WriteLine(obj1.Movie("Avengers") +" is my fav. movie");
16        }
17    }
```

Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\Desktop\Assignment 4\Day4A>dotnet run
i Study 2 hrs at night
Avengers is my fav. movie

C:\Users\hp\Desktop\Assignment 4\Day4A>

Enum

Source Code: -

```
using System;

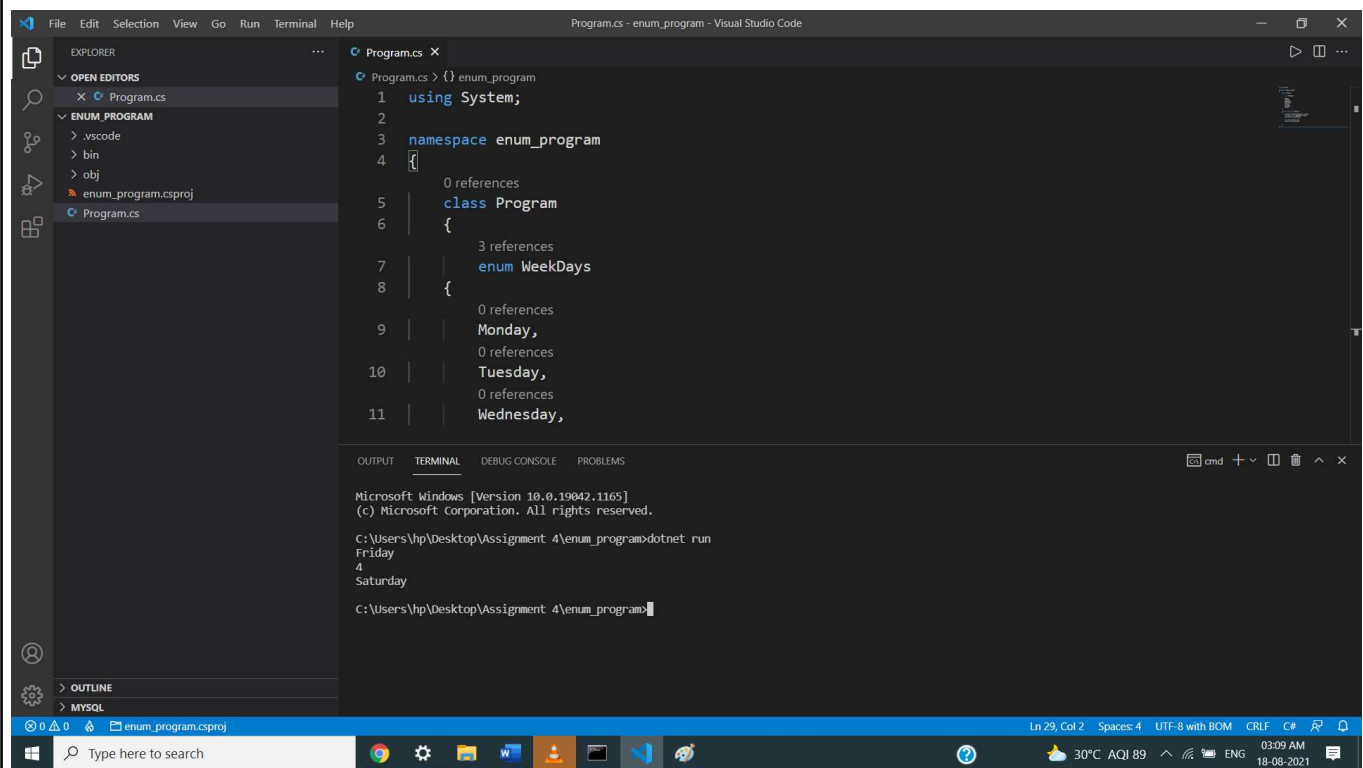
namespace enum_program
{
    class Program
    {
        enum WeekDays
        {
            Monday,
            Tuesday,
            Wednesday,
            Thursday,
            Friday,
            Saturday,
            Sunday
        }

        public static void Main()
        {
            Console.WriteLine(WeekDays.Friday);
            int day = (int) WeekDays.Friday;
            Console.WriteLine(day);

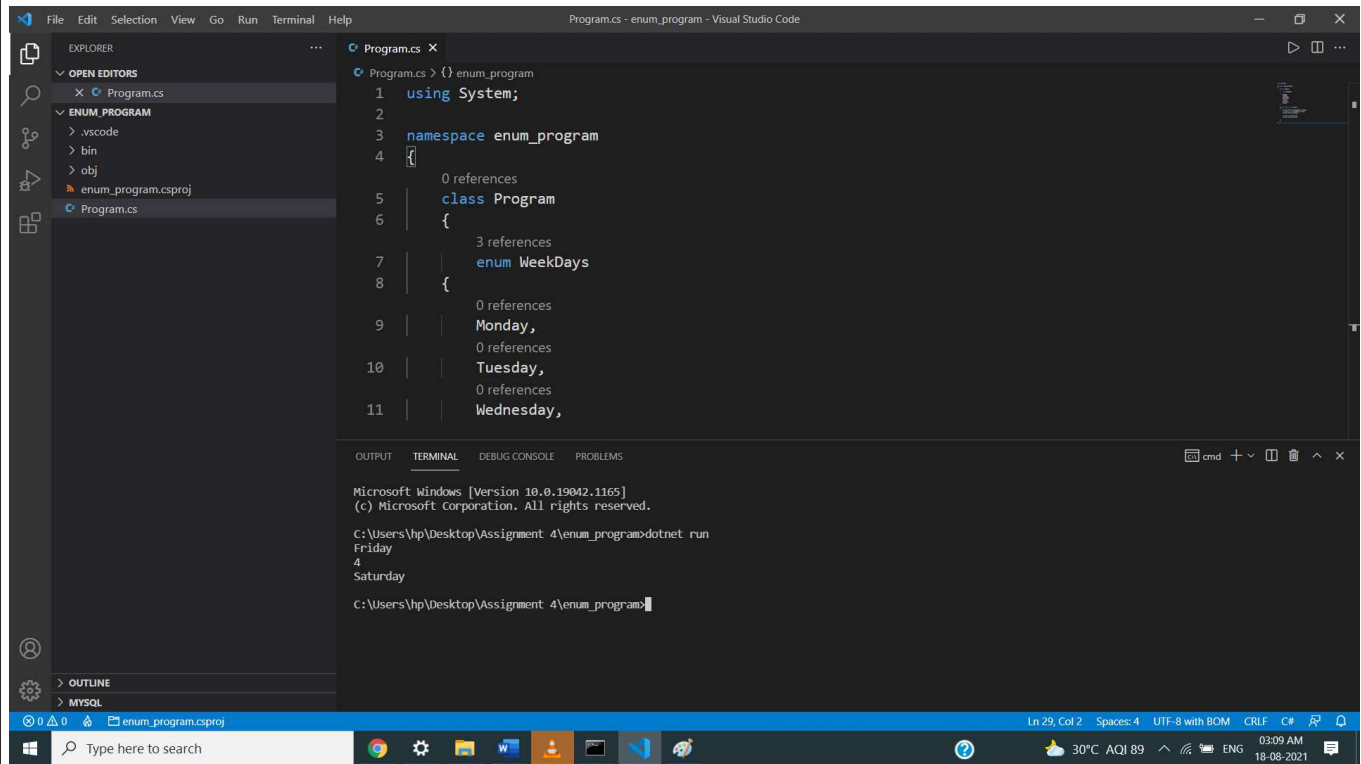
            var wd = (WeekDays)5;
            Console.WriteLine(wd);

        }
    }
}
```

Screen Shorts:-



Output :-



The screenshot displays the Visual Studio Code interface with a C# project named 'enum_program'. The Explorer sidebar on the left shows the project structure, including 'Program.cs'. The main editor window displays the code for 'Program.cs', which defines a namespace 'enum_program', a class 'Program', and an enum 'WeekDays' with values 'Monday', 'Tuesday', and 'Wednesday'. The code is as follows:

```
1 using System;
2
3 namespace enum_program
4 {
5     0 references
6     class Program
7     {
8         3 references
9         enum WeekDays
10        {
11            0 references
12            Monday,
13            0 references
14            Tuesday,
15            0 references
16            Wednesday,
```

Below the code editor, the TERMINAL panel shows the output of the command 'dotnet run'. The output displays the days of the week: 'Friday', 'Saturday', and 'Sunday'.

```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft corporation. All rights reserved.

C:\Users\hp\Desktop\Assignment 4\enum_program>dotnet run
Friday
Saturday
Sunday

C:\Users\hp\Desktop\Assignment 4\enum_program>
```

The status bar at the bottom indicates the file is 'enum_program.csproj', the encoding is 'UTF-8 with BOM', and the line/character count is 'Ln 29, Col 2'. The system tray shows the date and time as '03:09 AM 18-08-2021'.

Interface Program1

Source Code: -

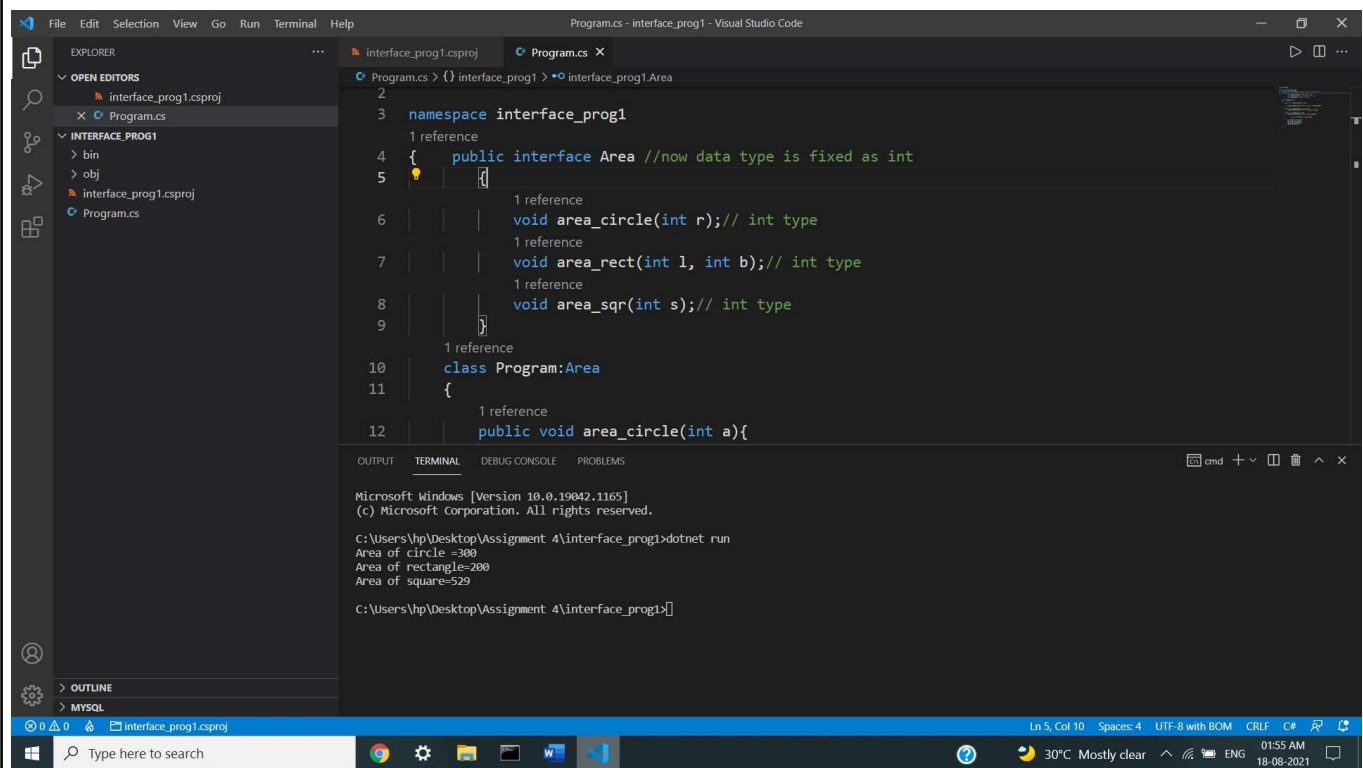
```
using System;

namespace interface_prog1
{
    public interface Area //now data type is fixed as int
    {
        void area_circle(int r); // int type
        void area_rect(int l, int b); // int type
        void area_sqr(int s); // int type
    }
    class Program:Area
    {
        public void area_circle(int a){

            Console.WriteLine("Area of circle =" + (22/7)*a*a);
        }
        public void area_rect(int a, int b){
            Console.WriteLine("Area of rectangle =" + a*b);
        }
        public void area_sqr(int a){
            Console.WriteLine("Area of square =" + a*a);
        }

        static void Main(string[] args)
        {
            var obj = new Program();
            obj.area_circle(10);
            obj.area_rect(10, 20);
            obj.area_sqr(23);
        }
    }
}
```

Screen Shorts:-



```
File Edit Selection View Go Run Terminal Help
Program.cs - interface_prog1 - Visual Studio Code

EXPLORER
  OPEN EDITORS
    interface_prog1.csproj
    Program.cs
  INTERFACE_PROG1
    bin
    obj
    interface_prog1.csproj
    Program.cs

Program.cs
2
3 namespace interface_prog1
4 {
5     public interface Area //now data type is fixed as int
6     {
7         void area_circle(int r); // int type
8         void area_rect(int l, int b); // int type
9         void area_sqr(int s); // int type
10    }
11    class Program:Area
12    {
13        public void area_circle(int a){
14
15        }
16    }
17 }
```

Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

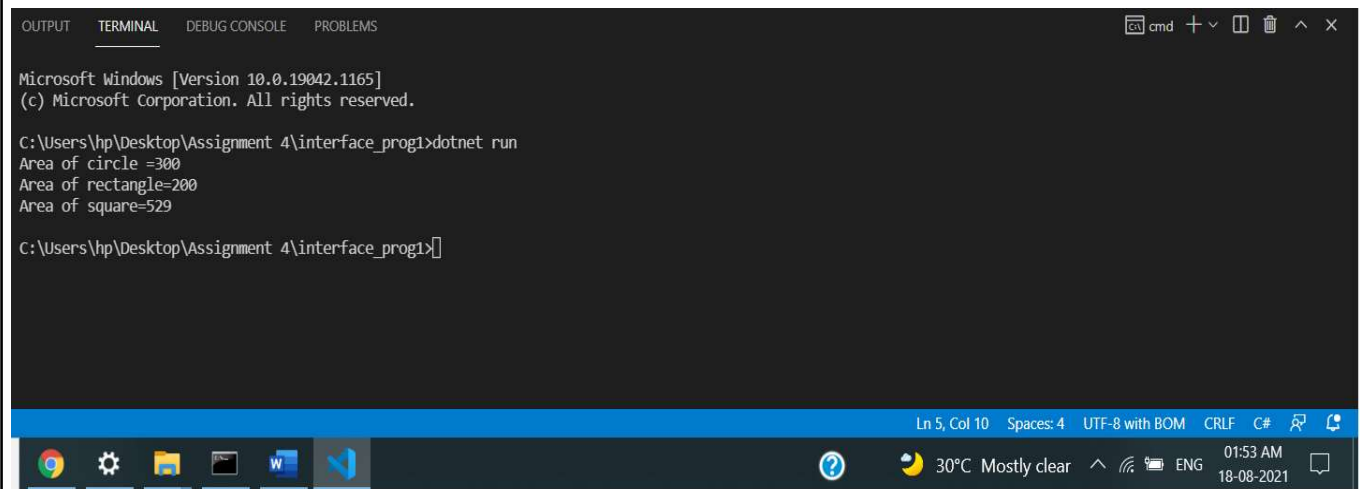
C:\Users\hp\Desktop\Assignment 4\interface_prog1>dotnet run
Area of circle =300
Area of rectangle=200
Area of square=529

C:\Users\hp\Desktop\Assignment 4\interface_prog1>

Ln 5, Col 10 Spaces: 4 UTF-8 with BOM CRLF C#

Type here to search 30°C Mostly clear 01:55 AM 18-08-2021

Output :-



The screenshot shows a Windows command prompt window with a dark background. The title bar at the top reads "cmd" and includes standard window controls. The terminal content is as follows:

```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\Desktop\Assignment 4\interface_prog1>dotnet run
Area of circle =300
Area of rectangle=200
Area of square=529

C:\Users\hp\Desktop\Assignment 4\interface_prog1>
```

At the bottom of the window, a status bar displays "Ln 5, Col 10", "Spaces: 4", "UTF-8 with BOM", "CRLF", "C#", and icons for search and refresh. Below the terminal window is the Windows taskbar, which includes icons for Google Chrome, Settings, File Explorer, Task View, Word, and Visual Studio. The system tray on the right shows a help icon, weather information ("30°C Mostly clear"), network and battery status, language ("ENG"), and the date and time ("01:53 AM 18-08-2021").

Interface Program2

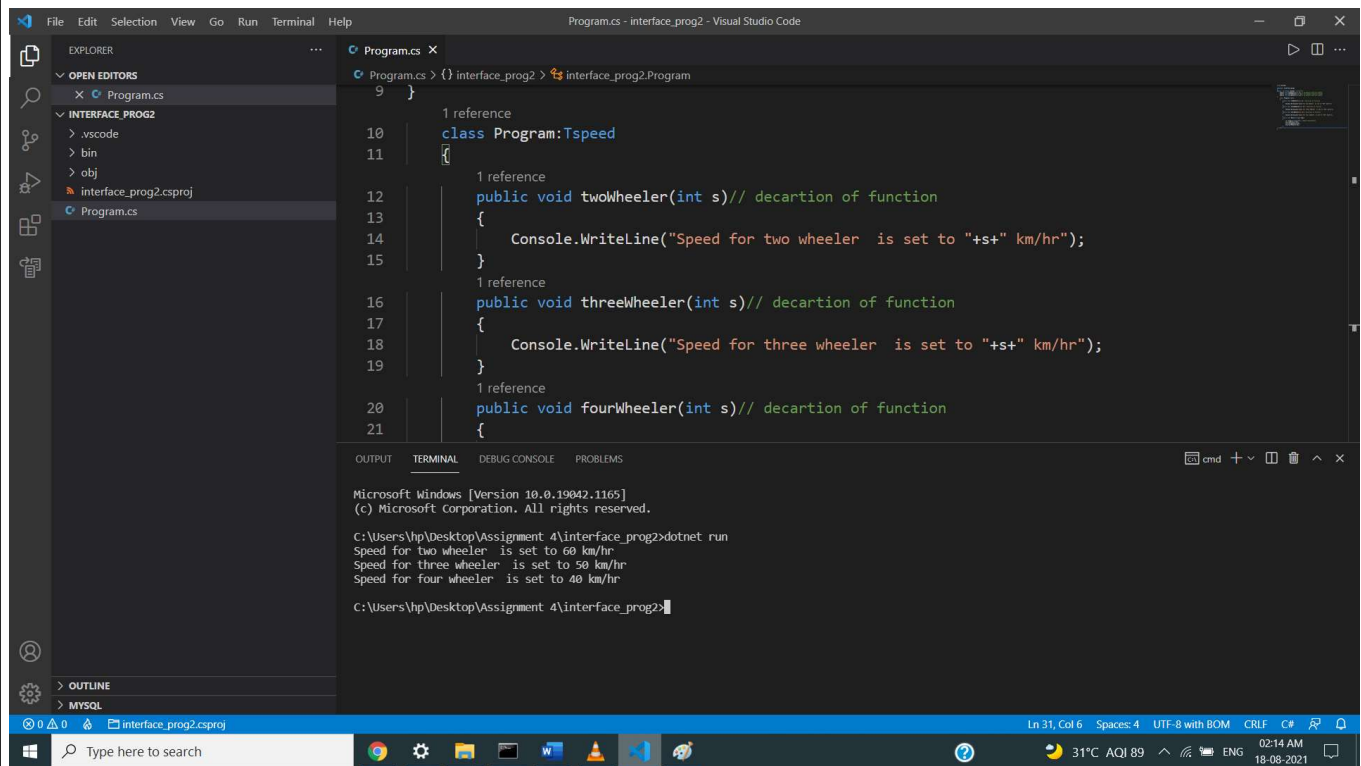
Source Code: -

```
using System;

namespace interface_prog2
{
    public interface Tspeed{ //interface
        public void twoWheeler(int a); // by default abstract method
        public void threeWheeler(int a); // by default abstract method
        public void fourWheeler(int a); // by default abstract method
    }

    class Program:Tspeed
    {
        public void twoWheeler(int s) // decartion of function
        {
            Console.WriteLine("Speed for two wheeler is set to "+s+" km/hr");
        }
        public void threeWheeler(int s) // decartion of function
        {
            Console.WriteLine("Speed for three wheeler is set to "+s+" km/hr");
        }
        public void fourWheeler(int s) // decartion of function
        {
            Console.WriteLine("Speed for four wheeler is set to "+s+" km/hr");
        }
        static void Main(string[] args)
        {
            var tsp=new Program(); //object declaration
            tsp.twoWheeler(60);
            tsp.threeWheeler(50);
            tsp.fourWheeler(40);
        }
    }
}
```

Screen Shorts:-



The screenshot displays the Visual Studio Code interface with a C# program open. The Explorer pane on the left shows the project structure, including the 'Program.cs' file. The main editor area shows the code for 'Program.cs', which defines a class 'Program' with three methods: 'twoWheeler', 'threeWheeler', and 'fourWheeler'. Each method takes an integer 's' as input and prints a message to the console. The output window at the bottom shows the results of running the program, displaying the speed for each vehicle type.

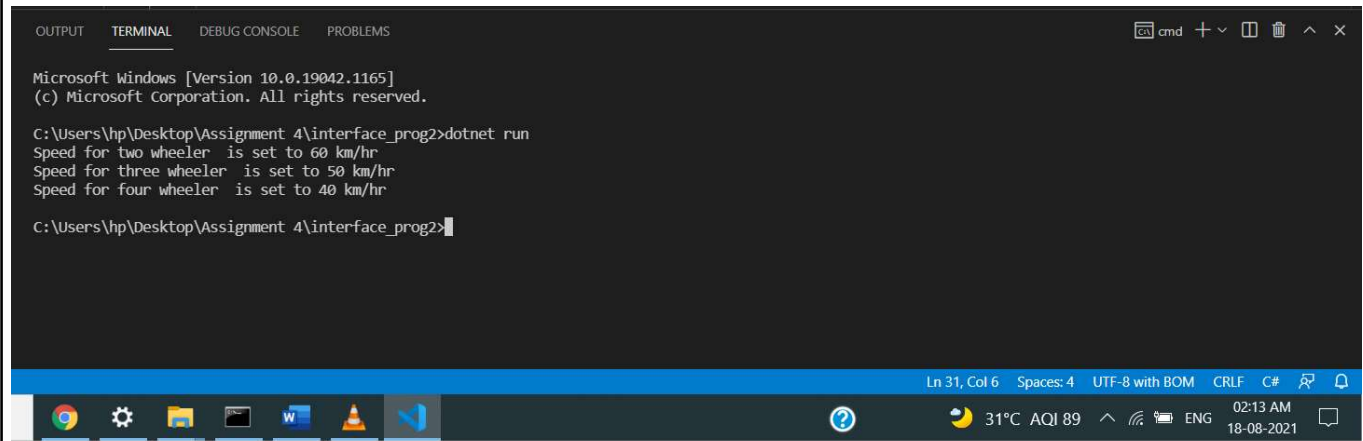
```
9 }
10
11     1 reference
    class Program:Tspeak
    {
        1 reference
        public void twoWheeler(int s)// decartion of function
        {
            Console.WriteLine("Speed for two wheeler  is set to "+s+" km/hr");
        }
        1 reference
        public void threeWheeler(int s)// decartion of function
        {
            Console.WriteLine("Speed for three wheeler  is set to "+s+" km/hr");
        }
        1 reference
        public void fourWheeler(int s)// decartion of function
        {
            Console.WriteLine("Speed for four wheeler  is set to "+s+" km/hr");
        }
    }
}
```

Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\Desktop\Assignment 4\interface_prog2>dotnet run
Speed for two wheeler is set to 60 km/hr
Speed for three wheeler is set to 50 km/hr
Speed for four wheeler is set to 40 km/hr

C:\Users\hp\Desktop\Assignment 4\interface_prog2>

Output :-



A screenshot of a Windows terminal window. The title bar at the top reads "cmd" with standard window controls. The terminal content shows the Windows version (10.0.19042.1165) and copyright information. It then displays the output of a .NET application run from the directory C:\Users\hp\Desktop\Assignment 4\interface_prog2. The application output lists speeds for two, three, and four wheelers. The terminal status bar at the bottom indicates the current line and column (Ln 31, Col 6), encoding (UTF-8 with BOM), and system information (31°C, AQI 89, 02:13 AM, 18-08-2021).

```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\Desktop\Assignment 4\interface_prog2>dotnet run
Speed for two wheeler  is set to 60 km/hr
Speed for three wheeler  is set to 50 km/hr
Speed for four wheeler  is set to 40 km/hr

C:\Users\hp\Desktop\Assignment 4\interface_prog2>
```

Ln 31, Col 6 Spaces: 4 UTF-8 with BOM CRLF C# 02:13 AM 18-08-2021

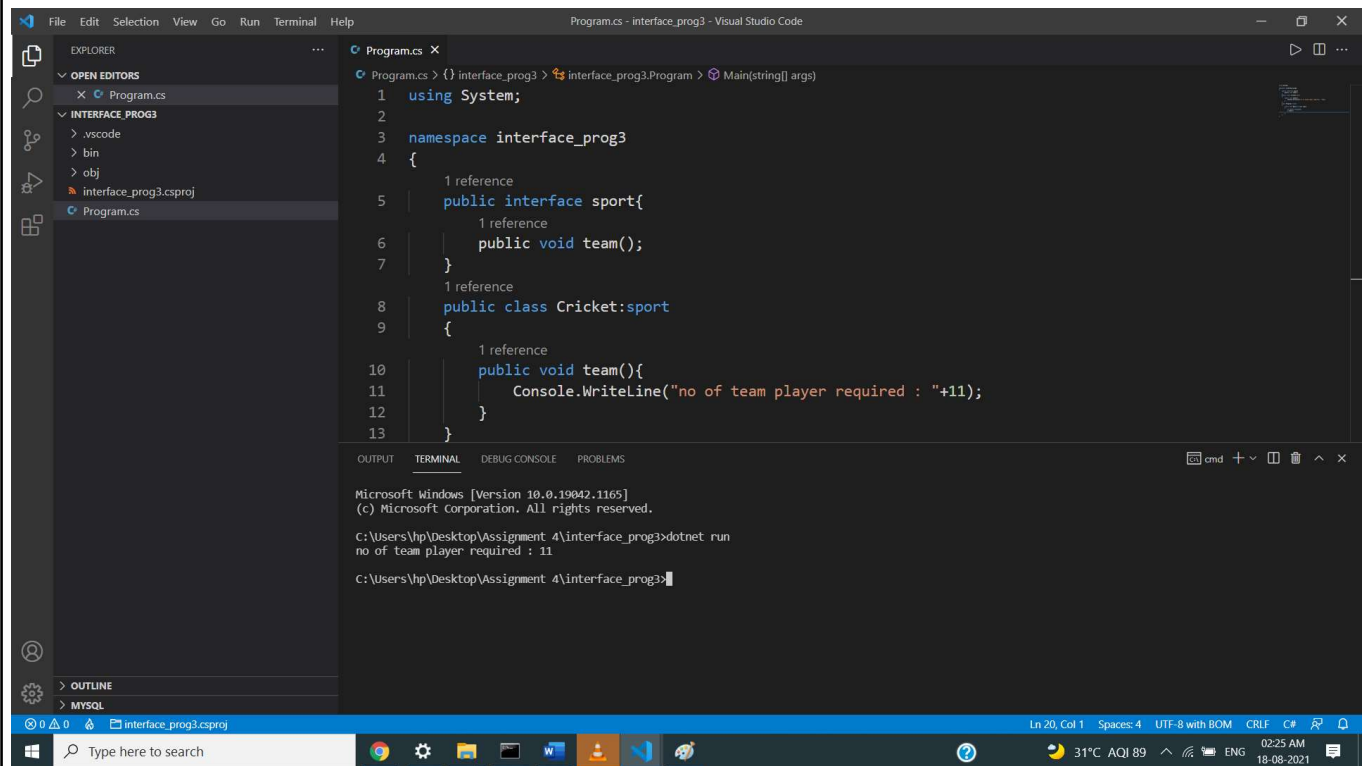
Interface Program3

Source Code: -

```
using System;

namespace interface_prog3
{
    public interface sport{
        public void team();
    }
    public class Cricket:sport
    {
        public void team(){
            Console.WriteLine("no of team player required : "+11);
        }
    }
    class Program:Cricket
    {
        static void Main(string[] args)
        {
            var t=new Program();
            t.team();
        }
    }
}
```


Screen Shorts:-



The screenshot displays the Visual Studio Code editor interface. The Explorer pane on the left shows the project structure with 'Program.cs' selected. The main editor area shows the following C# code:

```
1 using System;
2
3 namespace interface_prog3
4 {
5     1 reference
6     public interface sport{
7         1 reference
8         public void team();
9     }
10    1 reference
11    public class Cricket:sport
12    {
13        1 reference
14        public void team(){
15            Console.WriteLine("no of team player required : "+11);
16        }
17    }
18 }
```

The TERMINAL pane at the bottom shows the output of the program:

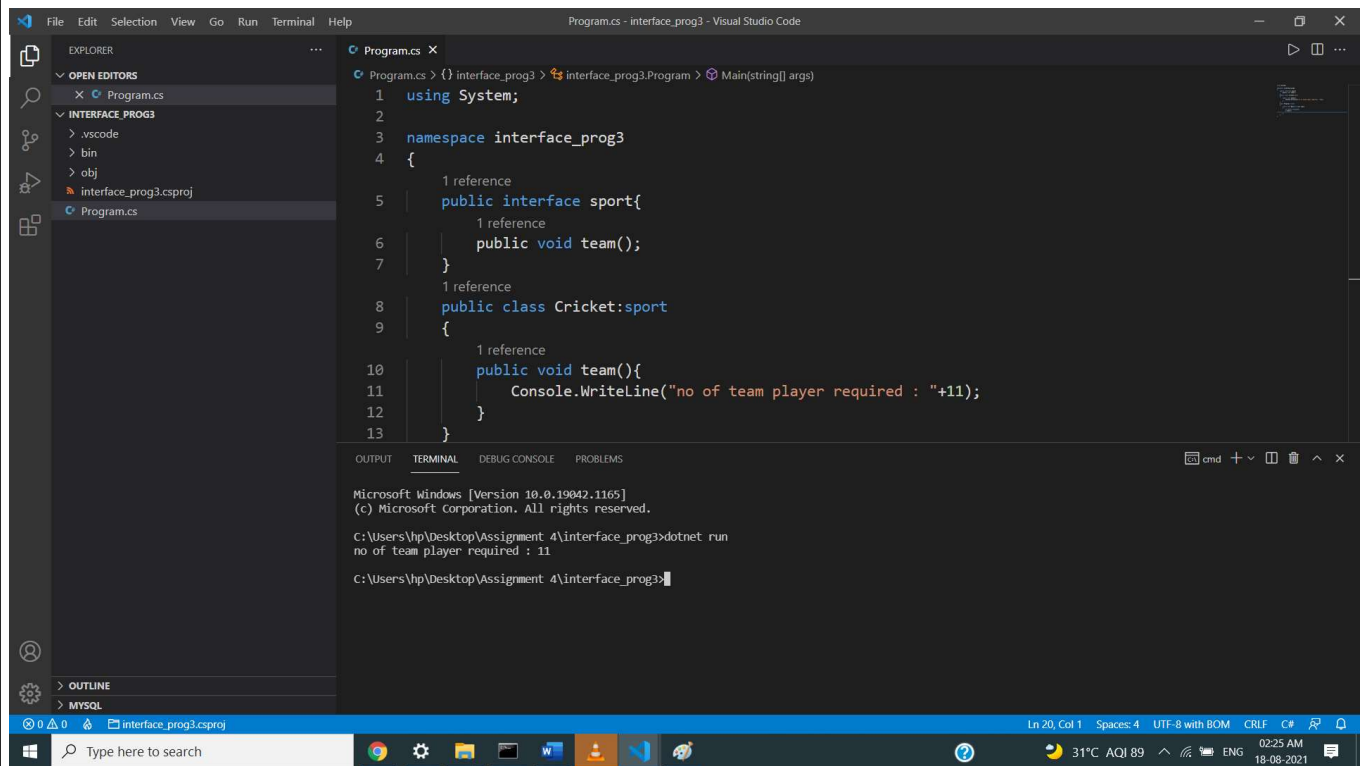
```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft corporation. All rights reserved.

C:\Users\hp\Desktop\Assignment 4\interface_prog3>dotnet run
no of team player required : 11

C:\Users\hp\Desktop\Assignment 4\interface_prog3>
```

The status bar at the bottom indicates the file is 'Program.cs', line 20, column 1, with 4 spaces, using UTF-8 encoding with BOM, CRLF line endings, and the C# language.

Output :-



The screenshot displays the Visual Studio Code interface with a C# program in the editor and its execution output in the terminal.

Program.cs

```
1 using System;
2
3 namespace interface_prog3
4 {
5     1 reference
6     public interface sport{
7         1 reference
8         public void team();
9     }
10    1 reference
11    public class Cricket:sport
12    {
13        1 reference
14        public void team(){
15            Console.WriteLine("no of team player required : "+11);
16        }
17    }
18 }
```

Terminal Output

```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\Desktop\Assignment 4\interface_prog3>dotnet run
no of team player required : 11

C:\Users\hp\Desktop\Assignment 4\interface_prog3>
```

The terminal output shows the program running successfully and printing "no of team player required : 11".

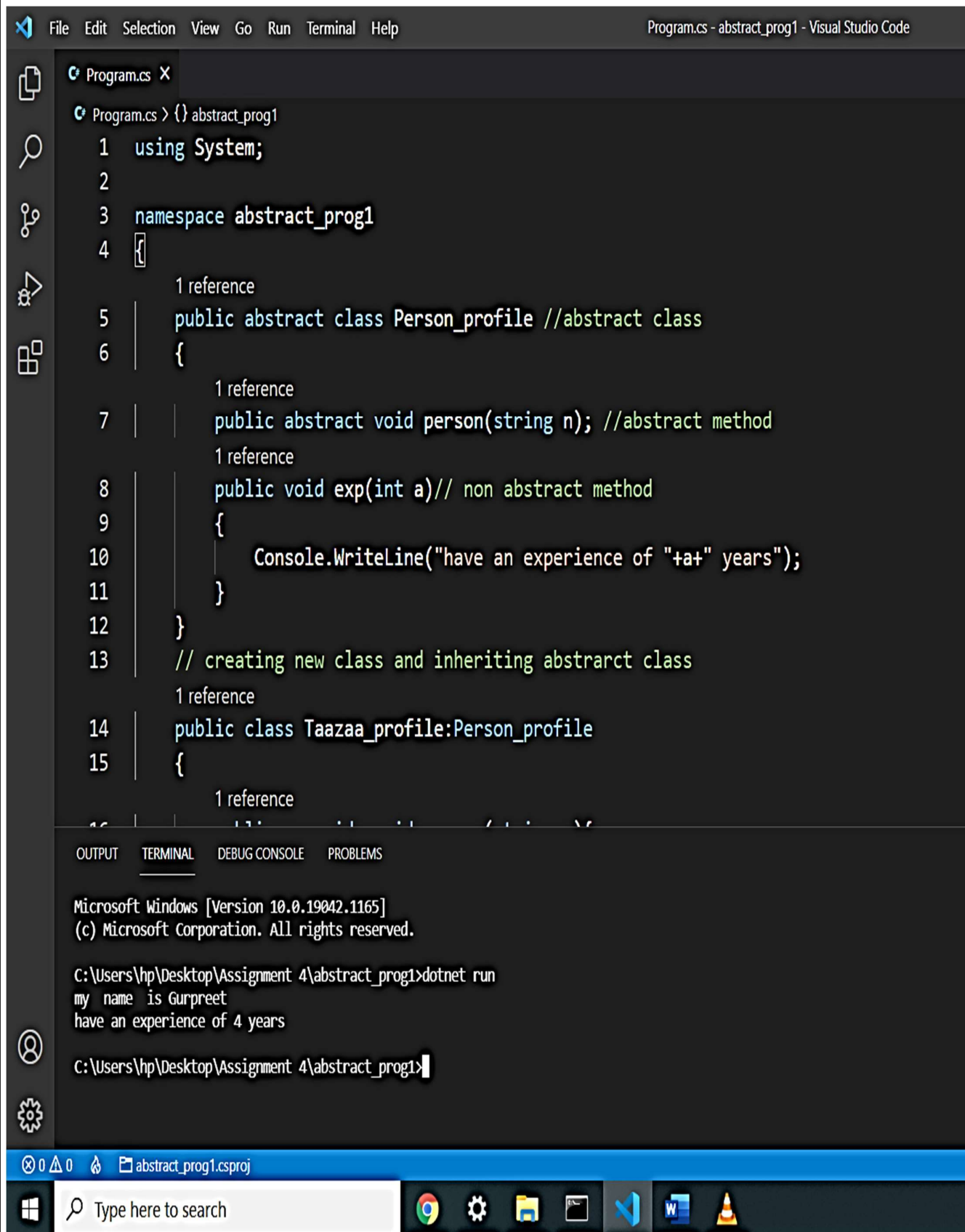
Abstract Class Program1

Source Code: -

```
using System;

namespace abstract_prog1
{
    public abstract class Person_profile //abstract class
    {
        public abstract void person(string n); //abstract method
        public void exp(int a)// non abstract method
        {
            Console.WriteLine("have an experience of "+a+" years");
        }
    }
    // creating new class and inheriting abstract class
    public class Taazaa_profile:Person_profile
    {
        public override void person(string n){
            Console.WriteLine("my name is "+n);
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            var obj=new Taazaa_profile();
            obj.person("Gurpreet");
            obj.exp(4);
        }
    }
}
```

Screen Shorts:-



The screenshot displays the Visual Studio Code interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar indicates the file is 'Program.cs - abstract_prog1 - Visual Studio Code'. The editor shows a C# file named 'Program.cs' with the following code:

```
1 using System;
2
3 namespace abstract_prog1
4 {
5     1 reference
6     public abstract class Person_profile //abstract class
7     {
8         1 reference
9         public abstract void person(string n); //abstract method
10        1 reference
11        public void exp(int a) // non abstract method
12        {
13            Console.WriteLine("have an experience of "+a+" years");
14        }
15    }
16    // creating new class and inheriting abstract class
17    1 reference
18    public class Taazaa_profile:Person_profile
19    {
20        1 reference
21    }
```

The bottom panel shows the 'TERMINAL' tab with the following output:

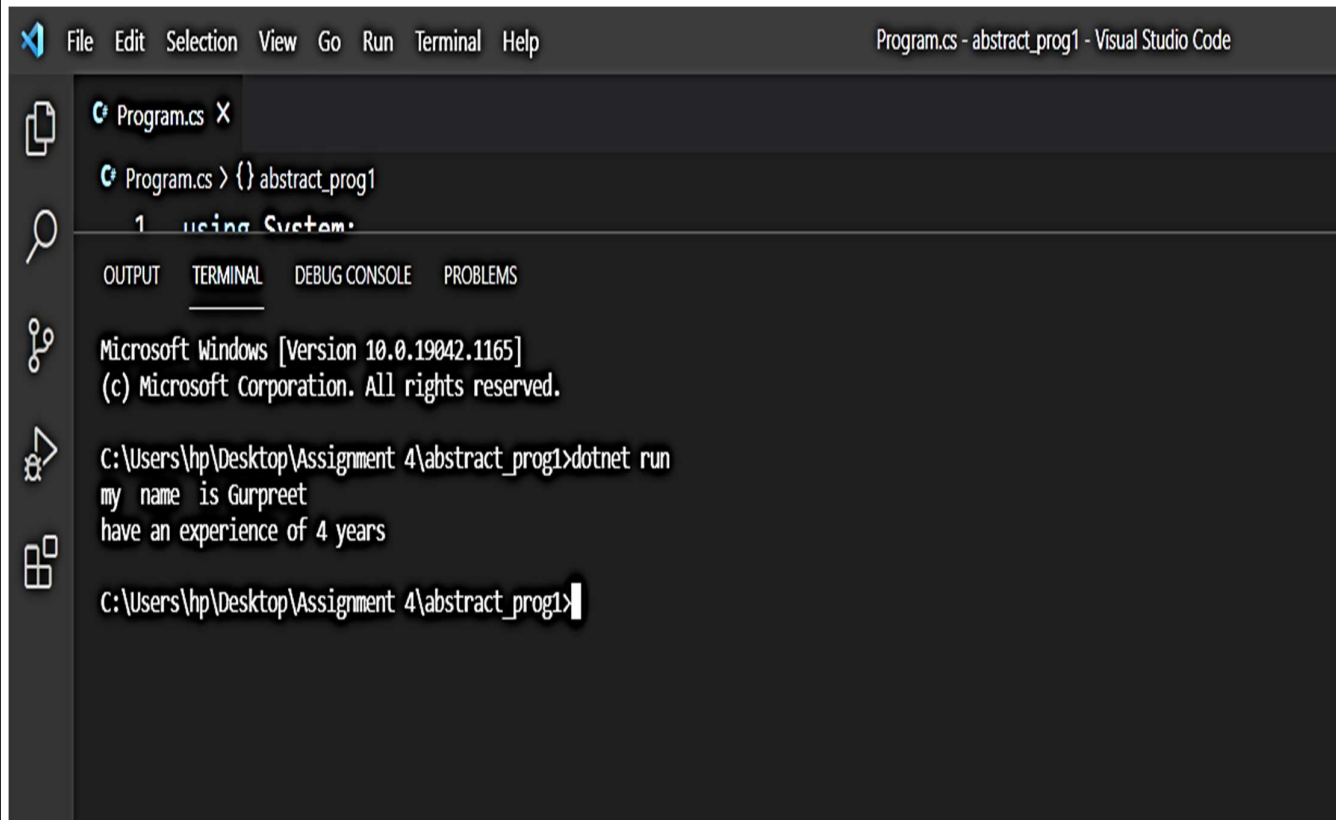
```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\Desktop\Assignment 4\abstract_prog1>dotnet run
my name is Gurpreet
have an experience of 4 years

C:\Users\hp\Desktop\Assignment 4\abstract_prog1>
```

The taskbar at the bottom shows the Windows Start button, a search bar, and several application icons including Chrome, Settings, File Explorer, Task View, Visual Studio Code, Word, and a game controller icon.

Output :-

A screenshot of the Visual Studio Code interface. The top menu bar shows 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', 'Terminal', and 'Help'. The title bar indicates the active file is 'Program.cs - abstract_prog1 - Visual Studio Code'. The editor area shows a C# file named 'Program.cs' with a single line of code: '1 using System;'. Below the editor, the 'TERMINAL' tab is selected, displaying the output of a command. The terminal text reads: 'Microsoft Windows [Version 10.0.19042.1165] (c) Microsoft Corporation. All rights reserved. C:\Users\hp\Desktop\Assignment 4\abstract_prog1>dotnet run my name is Gurpreet have an experience of 4 years C:\Users\hp\Desktop\Assignment 4\abstract_prog1>'.

```
File Edit Selection View Go Run Terminal Help
Program.cs - abstract_prog1 - Visual Studio Code

Program.cs X
Program.cs > {} abstract_prog1
1 using System;

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\Desktop\Assignment 4\abstract_prog1>dotnet run
my name is Gurpreet
have an experience of 4 years

C:\Users\hp\Desktop\Assignment 4\abstract_prog1>
```

Abstract Class Program2

Source Code: -

```
using System;

namespace abstract_prog2
{
    public abstract class Math{
        public abstract void classtime(int a);

    }
    class Program:Math
    {
        public override void classtime(int s)
        {
            Console.WriteLine("class time is "+s+" pm");
        }
        void scode(int b)
        {
            Console.WriteLine("subject code is :- "+b);
        }
        static void Main(string[] args)
        {
            var o=new Program();
            o.classtime(5);
            o.scode(101);
        }
    }
}
```

Screen Shorts:-

The screenshot displays the Visual Studio Code editor with a C# project named 'abstract_prog2'. The Explorer sidebar on the left shows the project structure, including 'Program.cs'. The main editor window shows the code for 'Program.cs' with the following content:

```
1 reference
5 | public abstract class Math{
6 |     1 reference
7 |     public abstract void classtime(int a);
8 | }
9 | 1 reference
10 | class Program:Math
11 | {
12 |     1 reference
13 |     public override void classtime(int s)
14 |     {
15 |         Console.WriteLine("class time is "+s+" pm");
16 |     }
17 |     1 reference
18 |     void scode(int b)
19 |     {
20 |         Console.WriteLine("subject code is :- "+b);
21 |     }
22 | 0 references
23 | static void Main(string[] args)
24 | {
25 |     var o=new Program();
```

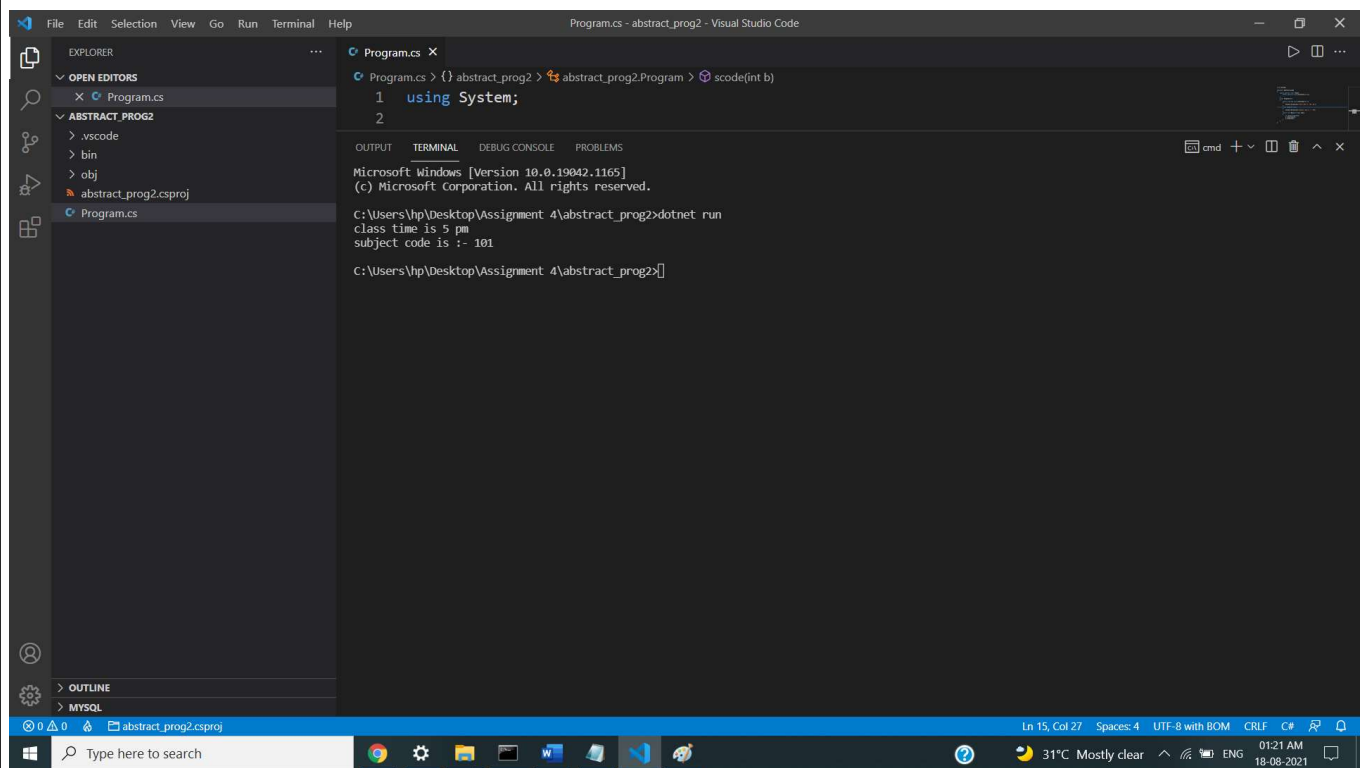
Below the code editor, the TERMINAL tab is active, showing the output of the program:

```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft corporation. All rights reserved.

C:\Users\hnp\Desktop\Assignment 4\abstract_prog2>dotnet run
class time is 5 pm
subject code is :- 101
```

The status bar at the bottom indicates the file is 'abstract_prog2.csproj', the cursor is at 'Ln 24, Col 10', and the encoding is 'UTF-8 with BOM'. The system tray shows the date and time as '01:20 AM 18-08-2021'.

Output :-



The screenshot displays the Visual Studio Code interface with a C# program named `Program.cs` open in the editor. The program contains the following code:

```
1 using System;  
2
```

The `OUTPUT` pane at the bottom shows the execution results of the command `dotnet run` in the directory `C:\Users\hp\Desktop\Assignment 4\abstract_prog2`. The output is as follows:

```
Microsoft Windows [Version 10.0.19042.1165]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\hp\Desktop\Assignment 4\abstract_prog2>dotnet run  
class time is 5 pm  
subject code is :- 101  
  
C:\Users\hp\Desktop\Assignment 4\abstract_prog2>
```

The status bar at the bottom indicates the current file is `abstract_prog2.csproj`, the cursor is at line 15, column 27, and the file is encoded in UTF-8 with BOM, using CRLF line endings.

Abstract Class Program3

Source Code: -

```
using System;

namespace abstract_prog3
{
    public abstract class RBI{//abstract class
        public abstract void accdetail(int acno,string name,string add,string actype);//abstract method
        public void minBalance(double a){
            Console.WriteLine("minimum balance required :- "+a);
        }
    }
    public class SBI:RBI
    {
        public override void accdetail(int acno, string name, string add, string actype)
        {
            Console.WriteLine("Acc no :- "+acno);
            Console.WriteLine("Acc Holder Name :- "+name);
            Console.WriteLine("Acc Type :- "+actype);
            Console.WriteLine("Acc Holder Address :- "+add);
        }
    }
    class Program:SBI
    {
        static void Main(string[] args)
        {
            var obj= new SBI();
            obj.accdetail(1576803,"Gurpreet Singh","New Delhi","Saving");
            obj.minBalance(3000);
        }
    }
}
```

Screen Shorts:-

The screenshot displays the Visual Studio Code interface with a C# project named 'abstract_prog3'. The Explorer pane on the left shows the project structure with files like .vscode, bin, obj, abstract_prog3.csproj, and Program.cs. The main editor shows the code for Program.cs, which defines an abstract class 'abstract_me' and a derived class 'SBI:RBI'.

```
6 | public abstract void accdetail(int acno,string name,string add,string actype);//abstract me
7 | 1 reference
8 | public void minBalance(double a){
9 |     Console.WriteLine("minimum balance required :- "+a);
10 | }
11 | 2 references
12 | public class SBI:RBI
13 | {
14 | 1 reference
15 |     public override void accdetail(int acno, string name, string add, string actype)
16 |     {
17 |         Console.WriteLine("Acc no :- "+acno);
18 |         Console.WriteLine("Acc Holder Name :- "+name);
19 |         Console.WriteLine("Acc Type :- "+actype);
20 |         Console.WriteLine("Acc Holder Address :- "+add);
21 |     }
22 | }
```

The TERMINAL pane at the bottom shows the output of the command 'dotnet run', displaying the account details for a user named Gurpreet Singh.

```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft corporation. All rights reserved.

C:\Users\hp\Desktop\Assignment 4\abstract_prog3>dotnet run
Acc no :- 1576803
Acc Holder Name :- Gurpreet Singh
Acc Type :- Saving
Acc Holder Address :- New Delhi
minimum balance required :- 3000

C:\Users\hp\Desktop\Assignment 4\abstract_prog3>
```

The status bar at the bottom indicates the current file is 'abstract_prog3.csproj', the cursor is at line 22, column 1, and the encoding is UTF-8 with BOM. The system tray shows the date and time as 01:43 AM on 18-08-2021.

Output :-

```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\Desktop\Assignment 4\abstract_prog3>dotnet run
Acc no :- 1576803
Acc Holder Name :- Gurpreet Singh
Acc Type :- Saving
Acc Holder Address :- New Delhi
minimum balance required :- 3000

C:\Users\hp\Desktop\Assignment 4\abstract_prog3>
```

Ln 22, Col 1 Spaces: 4 UTF-8 with BOM CRLF C#

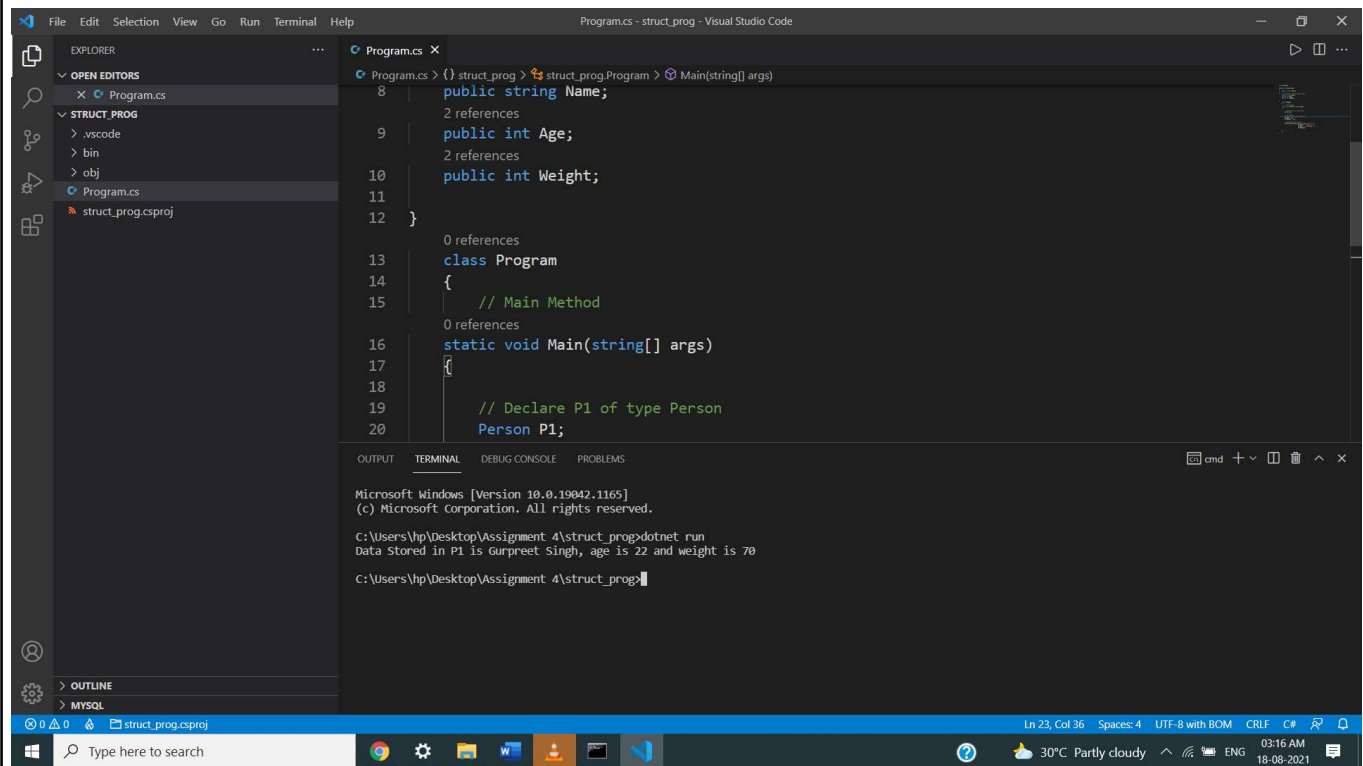
31°C Mostly clear 01:42 AM 18-08-2021

Struct Program

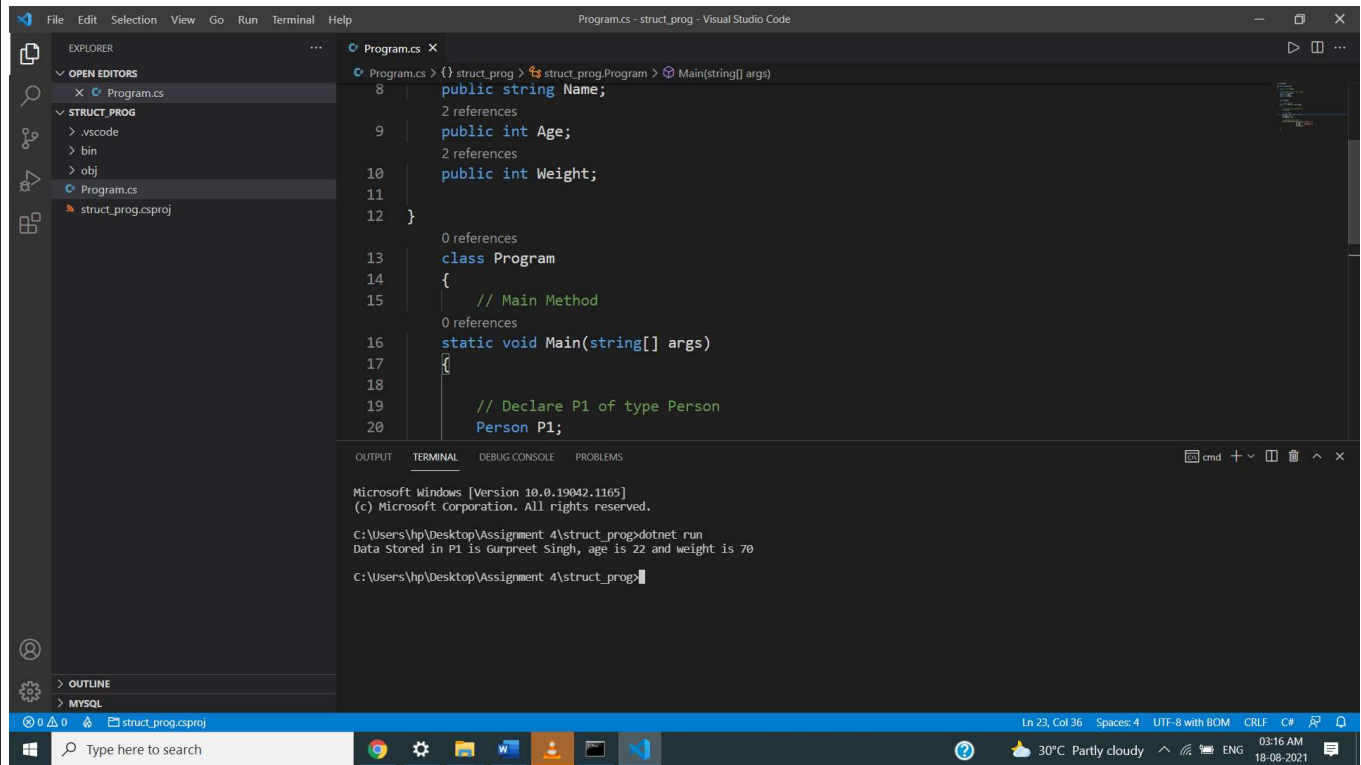
Source Code: -

[illegible]

Screen Shorts:-



Output :-



The screenshot displays the Visual Studio Code interface with a C# program named `Program.cs` open in the editor. The program defines a `Person` class with three public fields: `Name` (string), `Age` (int), and `Weight` (int). A `Program` class contains a `Main` method that declares a `Person` object `P1`. The terminal window shows the output of running the program, which prints the details of `P1`.

```
Program.cs > {} struct_prog > struct_prog.Program > Main(string[] args)
8 | public string Name;
  | 2 references
9 | public int Age;
  | 2 references
10 | public int Weight;
11 |
12 | }
  | 0 references
13 | class Program
14 | {
15 |     // Main Method
  | 0 references
16 | static void Main(string[] args)
17 | {
18 |
19 |     // Declare P1 of type Person
20 |     Person P1;
```

Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\Desktop\Assignment 4\struct_prog>dotnet run
Data Stored in P1 is Gurpreet Singh, age is 22 and weight is 70

C:\Users\hp\Desktop\Assignment 4\struct_prog>

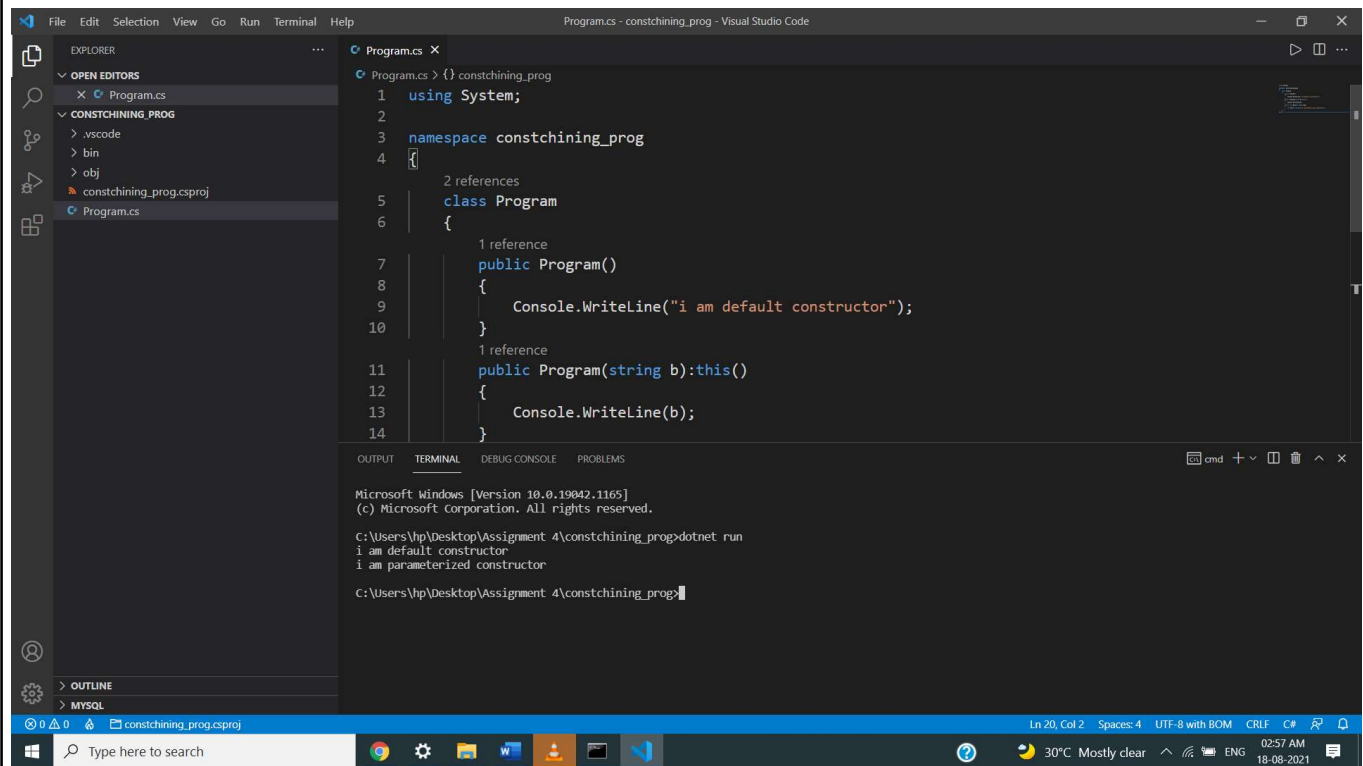
Constructor Chaining

Source Code: -

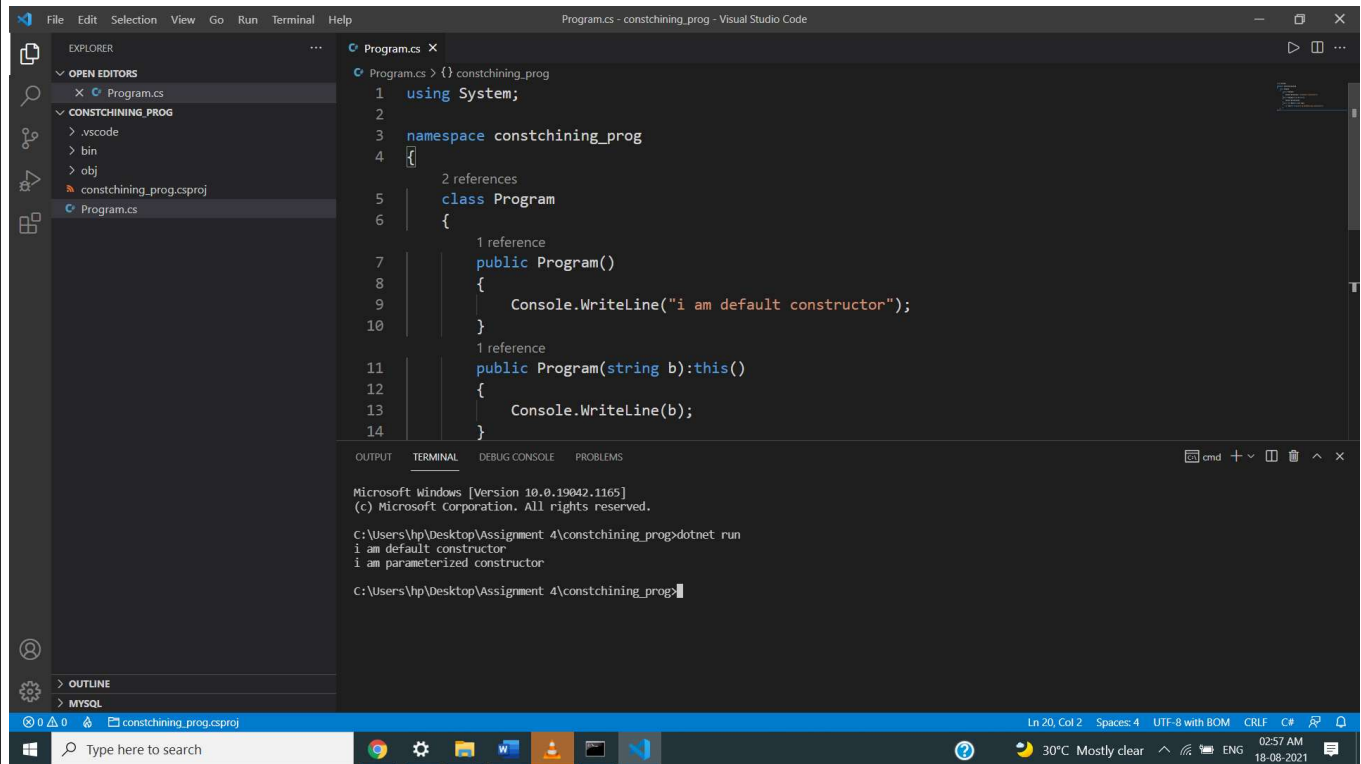
```
using System;

namespace constchining_prog
{
    class Program
    {
        public Program()
        {
            Console.WriteLine("i am default constructor");
        }
        public Program(string b):this()
        {
            Console.WriteLine(b);
        }
        static void Main(string[] args)
        {
            var obj=new Program("i am parameterized constructor");
        }
    }
}
```

Screen Shorts:-



Output :-



The screenshot displays the Visual Studio Code interface with a C# program named `Program.cs` open in the editor. The program defines a `Program` class with two constructors: a default constructor and a parameterized constructor. The output window shows the results of running the program, confirming that both constructors are executed successfully.

```
Program.cs > {} constchining_prog
1 using System;
2
3 namespace constchining_prog
4 {
5     2 references
6     class Program
7     {
8         1 reference
9         public Program()
10        {
11            Console.WriteLine("i am default constructor");
12        }
13        1 reference
14        public Program(string b):this()
15        {
16            Console.WriteLine(b);
17        }
18    }
19 }
```

OUTPUT

```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft corporation. All rights reserved.

C:\Users\hp\Desktop\Assignment 4\constchining_prog>dotnet run
i am default constructor
i am parameterized constructor

C:\Users\hp\Desktop\Assignment 4\constchining_prog>
```