|  |
| --- |
| **Day-14 Assignment**  **By**  **Bhanu Rama Krishna Prakash Jakkamsetti**  **10/2/2022** |

|  |
| --- |
| 1.Research and write what is the use of sealed class.  WACP to illustrate sealed class. |
| Sealed class is used to stop a class to be inherited. You can’t derive or extend any class from it. Sealed method is implemented so that no other class can overthrow it and implement its own class. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day14\_project1  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author:bhanu rama krishna prakash jakkamsetti  \* Purpose:creating sealed class  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  sealed class Police  {  public static int EmgNumber = 100;  public string TakeAction()  {  return "Tke Action immidiatly";  }  }  internal class Program  {  static void Main(string[] args)  {  Police p = new Police();  Console.WriteLine(Police.EmgNumber);  Console.WriteLine(p.TakeAction());  }  }  } |
| Output: |
|  |

|  |
| --- |
| 2.reasearch and write what is the difference between normal and auto-implemented properties. |
| * Read-Only Properties: When property contains only get method. * Write Only Properties: When property contains only set method. * Auto Implemented Properties: When there is no additional logic in the property accessors and it introduce in C# 3.0. |

|  |
| --- |
| WACP to illustrate normal properties. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day14\_project2  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author:bhanu rama krishna prakash jakkamsetti  \* Purpose:initializing normal properties  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  class Averagespeed  {  private int time;  private int distance;  private int speed;  public int Time  {  set { time = value; }  }  public int Distance  {  set { distance = value; }  }  public int Speed  {  get  {  speed = distance / time;  return speed;  }  }  internal class Program  {  static void Main(string[] args)  {  Averagespeed a=new Averagespeed();  a.Time = 30;  a.Distance = 120;  Console.WriteLine(a.Speed);  }  }  }  } |
| Output: |
|  |

|  |
| --- |
| WACP to illustrate Auto-implemented properties. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day14\_project3  {/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author:bhanu rama krishna prakash jakkamsetti  \* Purpose:initializing auto implemented properties  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  class Averagespeed  {  private int time;  private int distance;    public int Time  {  set { time = value; }  }  public int Distance  {  set { distance = value; }  }  public int Speed  {  get  {  return distance / time;    }  }  internal class Program  {  static void Main(string[] args)  {  Averagespeed a = new Averagespeed();  a.Time = 30;  a.Distance = 120;  Console.WriteLine(a.Speed);  }  }  }  } |
| Output: |
|  |

3. Research and fix the below issue:

interface IRules

{

int Age { get; set; }

int add(int a, int b);

public void PrintHi()

{

Console.WriteLine("Hi");

}

}

3. Research and fix the below issue:

interface IRules

{

int Age { get; set; }

int add(int a, int b);

public void PrintHi()

{

Console.WriteLine("Hi");

}

}

|  |
| --- |
| 3.Research and fix the below issue:  Interface Irules  {  Int age{get; set;}  Int add(int a, int b);  Public void PrintHi()  {  Console.writeLine(“hiii”);  }  } |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day14\_project7  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author:bhanu rama krishna prakash jakkamsetti  \* purpose:declaring normal method in interface  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  interface IStudent  {  int Id { get; set; }  string Name { get; set; }  public void PrintHi();    }  class Lol : IStudent  {  public void PrintHi()  {  Console.WriteLine("hiiiiii");  }  }  internal class Program  {  static void Main(string[] args)  {    }  }  } |
| Output: |
|  |

|  |
| --- |
| 4.WACP to check if the number is prime or not using break logic discussed in the class. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day14\_project4  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author:bhanu rama krishna prakash jakkamsetti  \* purpose:print prime by using break  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  internal class Program  {  static void Main(string[] args)  {  int n = 78, i;  for ( i = 2; i < n; i++)  {  if (n % i == 0)  break;  }  if (i == n)  {  Console.WriteLine("is prime");  }  else  {  Console.WriteLine("not prime");  }  }  }  } |
| Output: |
|  |

|  |
| --- |
| 5.Print numbers from 1-30 and skip the numbers divisible by 3 using continue. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day14\_project5  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author:bhanu rama krishna prakash jakkamsetti  \* purpose:use continue keyword  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  internal class Program  {  static void Main(string[] args)  {  for (int i = 1; i <=30; i++)  {  if (i%3==0)  continue;  Console.Write($"{i} ");  }  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 6.Find the first number after 1000 which is divisible by 97 use for loop and break. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day14\_project6  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author:bhanu rama krishna prakash jakkamsetti  \* purpose:first number after 1000 which is divisible by 97  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  internal class Program  {  static void Main(string[] args)  {  for (int i =1000; i <= 1097; i++)  {  if (i % 97 == 0)  {  Console.WriteLine(i);  break;  }  }  Console.ReadLine();  }  }  } |
| Output: |
|  |