|  |
| --- |
| Day 14 Assignment  By  B.P.N.V.S.Sudheer  10-02-22 |

|  |
| --- |
| 1.Research and write what is the use of sealed class.WACP to  Illustrate sealed class |
| * Sealed class is used ton stop a class to be inherited you cannot derive or extend any class from it sealed method is implemented so that no other class can overthrow it and implement its own method |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace Day14\_project1  {  sealed class police  {  public static int helpline = 100;  public string getpersonalphno()  {  return "9398388479";  }  }  class theif  {  public string name()  {  return "sai";  }  }          internal class Program  {    static void Main(string[] args)  {  Console.WriteLine(police.helpline);      police p = new police();    Console.WriteLine(p.getpersonalphno());  theif t = new theif();  Console.WriteLine(t.name());  Console.ReadLine();    }  } |
| Output: |
|  |
| 2.WACP to check if the number is prime or not using logic discussed in the class class hint:use break |
| Code : |

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace \_14thdayproject2  {  internal class Program  {  static void Main(string[] args)  {  int n = 96, i;  for (i = 2; i < n; i++)  {  if (n % i == 0)  break;  }  if (i == n)  Console.WriteLine("prime");  else  Console.WriteLine("not prime");  Console.ReadLine();  }  }  } |
| Output: |
|  |
| 3.Print numbers from 1 to 30 and skip the numbers divisible by 3 hint:use continue |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace \_14thdayproject3  {  internal class Program  {  static void Main(string[] args)  {  for ( int i=0; i<=30; i++)  {  if (i % 3 == 0)  continue;  Console.WriteLine(i);    }  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 4.Find the first number after 1000 which is divisible by 97 using for loop and break |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace \_14thdayproject4  {  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: |
|  |
| 5.Research and write about normal properties and auto implemented property |
| Auto implemented property: |
| * A property in a class without explicity providing the get and set accessors * For an implementated property the compiler automatically creates a private field to store the property variable the corresponding get and set accessors |

|  |
| --- |
| Normal Property   * A property is a member that provides a flexible mechanism to read,write,or compute the value of a private field properties can be used as if they are public data member but they are actually special methods called accessors |
| 6.WACP To illustrate normal propertis |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace \_14thdayproject5  {  class student  {  private string name;  public string Name  {  get  {  return name;  }  set  {  name = value;  }  }  }  internal class Program  {  static void Main(string[] args)  {  student s = new student();  s.Name = "sai";  Console.WriteLine(s.Name);  Console.ReadLine();    }    }  } |
| Output: |
|  |
| 7.WACP to illustrate auto implemented property |
| Code: |

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace \_14thdayproject6  {  internal class Program  {  class student  {  public string name { get; set; }  }  static void Main(string[] args)  {  student s = new student();  s.name = "sudheer";  Console.WriteLine(s.name);  Console.ReadLine();  }  }  } |
| Output: |
|  |
|  |
|  |
|  |
|  |
|  |