|  |
| --- |
| DAY14 ASSIGNMENT  BY  PALURU MOUNIKA  10-02-2022 |

|  |
| --- |
| **1.What is the use of sealed class and write a c# program to illustrates sealed class?** |
| **Use of sealed class:**  1.sealed class is used to stop a class to be inherited. |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Author:paluru mounika  //purpose:program to iilustrate sealed methode  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day14project1  {  sealed class Police  {  public static int Helpline = 100;  public string Getsecret()  {  return "007";  }  }  internal class Program  {  static void Main(string[] args)  {  Police p = new Police();  Console.WriteLine(p.Getsecret());  Console.WriteLine(Police.Helpline);  Console.ReadLine();  }  }  } |
| **Output:** |
|  |

|  |  |
| --- | --- |
| **2.What is the difference between normal property and auto implemented property**  **a)write a c# program auto implemented property**  **b)write a c# program normal property.** | |
| **Auto implemented properties** | **Normal properties** |
| **1.It must consists of get;method and set; is optional**  **2.it is a member that provides a flexible mechanism for classes to expose private fields.** | **1.these are similar to class variables with get; and set; methodes**  **2.it enables you to quickly specify a property of a class without having to write code to get and set the property.** |
| **Auto-implimented method:** | |
| **Code:** using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace day14project2  {  class Student  {  public int Id { get; set; }  public string Name { get; set; }  public string Mail { get; set; }  }  internal class Program  {  static void Main(string[] args)  {  Student student = new Student();  student.Id = 101;  student.Name = "mounika";  student.Mail = "mouni@gmail.com";  Console.WriteLine(student.Id);  Console.WriteLine(student.Name);  Console.WriteLine(student.Mail);  Console.ReadLine();  }  }  } | |
| **Output:** | |
|  | |
| **Normal properties:** | |
| **Code:** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Author:paluru mounika  //purpose:program to illustrate the normal property  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day14project2b  {  class Student  {  private int id;  private string name;  private string mail;  public int Id  {  get { return id; }  set { id = 101; }  }  public string Name  {  get { return name; }  set { name = "pavani"; }  }  public string Mail  {  get { return mail; }  set { mail = "pavani@gmail.com"; }  }    }  internal class Program  {  static void Main(string[] args)  {  Student student = new Student();  student.Id = 101;  student.Name = "pavani";  student.Mail = "pavani@gmail.com";  Console.WriteLine(student.Id);  Console.WriteLine(student.Name);  Console.WriteLine(student.Mail);  Console.ReadLine();  }  }  } | |
| **Output:** | |
|  | |

|  |
| --- |
| **4.write a c# program check the number is prime or not use cointinue.** |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Author:paluru mounika  //Purpose:prime number or not using break  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day14project4  {  internal class Program  {  static void Main(string[] args)  {  int n = 7, i;  for (i = 2; i < n; i++)  {  if (n % i == 0)  break;  }  if (i == n)  Console.WriteLine("prime number");  else  Console.WriteLine("not a prime number");  Console.ReadLine();  }  }  } |
| **Output:** |
|  |

|  |
| --- |
| **5.print numbers from 1 to 30 and skip the numbers divisible by 3**  **Hint:use forloop and break.** |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Author:paluru mounika  //Purpose:print 1 to 30 numbers using continue  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day14project5  {  internal class Program  {  static void Main(string[] args)  {  for (int i = 1; i <= 30; i++)  {  if (i % 3 == 0)  continue;  Console.WriteLine(i);  }  Console.ReadLine();  }  }  } |
| **Output:** |
|  |

|  |
| --- |
| **6.Find the first nuber after 1000 which is divisible by 97.** |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Author:paluru mounika  //Purpose:first number after 1000 divisible by 97  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day14project6  {  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:** |
|  |
|  |