|  |
| --- |
| Day-14 Morning Assignment  By  U.Joshna  [10-2-2022] |

|  |
| --- |
| 1.Research and write what is the use of Sealed Class.  WACP to illustrate Sealed Class. |
| Use of Sealed Class: |
| .Sealed Class is used to 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 Sealed\_Class  {  sealed class SealedClass  {  public int Add(int a, int b)  {  return a + b;  }  }  internal class Program  {  static void Main(string[] args)  {  SealedClass slc = new SealedClass();  int total = slc.Add(10, 20);  Console.WriteLine("Total = " + total.ToString());  Console.ReadLine();  }  }  } |
| Output: |
|  |
|  |

|  |
| --- |
| 2. Research and write what is the difference  between normal properties and auto-implemented  properties.  A) WACP to illustrate normal properties  B) WACP to issustrate auto-implemented properties |
| Normal Properties: |
| .Properties expose fields.Fields should ( almost always) be kept private to a class and accessed via get and set properties. Properties provide a level of abstraction allowing you to change the fields while not affecting the external way they are accessed  By the things that use your class |
| .Auto-Implemented Properties: |
| In C# 3.0 and later , auto implemented properties make property –declaration more concise when no additional logic is required in the property accessors. They also enable client code to create objects .In C# 9 and later , init accessors can also be declared as  Auto-Implemented properties |
| Code A: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Normal\_Properties  {  class Employee  {  private int id;  private string name;  private string designation;  private int salary;  public int Id  {  get { return id; }  set { id = value; }  }  public string Name  {  get { return name; }  set { name = value; }  }  public string Designation  {  set { designation = value; }  }  public int Salary  {  get  {  salary = (designation == "N") ? 20000 : 300000;  return salary;  }  }  }  internal class Program  {  static void Main(string[] args)  {  Employee emp = new Employee();  emp.Designation = "Manager";  Console.WriteLine(emp.Salary);  Console.ReadLine();  }  }  } |
| Output: |
|  |
| Code B: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Auto\_Implemented\_Properties  {  class Student  {    public int ID { get; set; }  public string Name { get; set; }  public string Email { get; set; }  }  internal class Program  {  static void Main(string[] args)  {  Student student = new Student();    student.ID = 103;  student.Name = "JOSHNA U";  student.Email = "josh@77.com";    Console.WriteLine(student.ID);  Console.WriteLine(student.Name);  Console.WriteLine(student.Email);  Console.ReadLine();  }  }  } |
| 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");  }  } |
| Code: |
|  |
|  |
|  |

|  |
| --- |
| 4. WACP to check if the number is prime or not  using logic discussed in the class  HINT : use break; |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Prime\_or\_Not  {  internal class Program  {  static void Main(string[] args)  {  int n = 17, i;  for (i = 2; i < n; i++)  {  if (n % i == 0)  break;  }  if (i == n)  {  Console.WriteLine("{0} is Prime Number", n);  }  else  {  Console.WriteLine("{0} is not a Prime Number", n);  }  Console.ReadLine();  }  }  } |
| Output: |
|  |
|  |
| 5. 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 Day\_14\_Project\_5  {  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 number after 1000 which is  divisible by 97.  HINT : use for loop and break |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day\_14\_project\_6  {  internal class Program  {  static void Main(string[] args)  {  for (int i = 1000; i <= 1095; i++)  {  if (i % 97 == 0)  {  Console.WriteLine(i);  break;  }  }  Console.ReadLine();  }  }  } |
| Output: |
|  |
|  |
|  |
|  |