|  |
| --- |
| Day 14(10 feb) Assignment  By Ramakrishna |

|  |
| --- |
| **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. In other word, you cannot derive or extend any class from it. Secondly, Sealed method is used, so that no other class can override 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 police  {  public static int helpline = 101;  public string GetSecret()  {  return "222" ;  }  }  internal class Program  {  static void Main(string[] args)  {  police p = new police();  Console.WriteLine( p.GetSecret());  Console.WriteLine(police.helpline);  Console.ReadLine();  }  }  } |
| Output : |
|  |

|  |
| --- |
| **2 .WACP to illustrate normal properties** |
| 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 members, but they are actually special methods called accessors. |
| Code: |
| 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 int salary;  public int Id  {  get { return id; }  set { id = 5555; }  }  public string Name  {  get { return name; }  set { name = "Rk"; }  }  public int Salary  {  get { return salary; }  set { salary = 10000; }  }  }  internal class Program  {  static void Main(string[] args)  {  Employee emp = new Employee();  emp.Id = 5555;  emp.Name = " Rk";  emp.Salary = 10000;  Console.WriteLine(emp.Id);  Console.WriteLine(emp.Name);  Console.WriteLine(emp.Salary);  Console.ReadLine();    }  }  } |
| Output: |
|  |

|  |
| --- |
| **3 . WACP to issustrate auto-implemented properties** |
| Auto-implemented properties enable you to quickly specify a property of a class without having to write code to Get and Set the property |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace auto\_implemented\_properties  {  class Employee  {  public int Id { get; set; }  public string Name { get; set; }  public int Salary { get; set; }  }  internal class Program  {  static void Main(string[] args)  {  Employee emp = new Employee();  emp.Id = 9999;  emp.Name =" RAMA KRISHNA ";  emp.Salary = 250000;  Console.WriteLine(emp.Id);  Console.WriteLine(emp.Name);  Console.WriteLine(emp.Salary);  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| **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\_number  {  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("NUmber is not prime");  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 printing\_\_numbers  {  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.**  **HINT : use for loop and break.** |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace find\_the\_first\_number  {  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: |
|  |