|  |
| --- |
| **Day14 Morning Assignments**  **By**  **Manoj Yekolla**  **10-Feb-2022** |

|  |
| --- |
| **1(a)WACP to illustrate sealed class.** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace SealedClass  {  sealed class Police  {  public static int code = 100; //static variable  /\* public string GetSecret() //method  {  return "003";  }\*/  }  internal class Program  {  static void Main(string[] args)  {  // Police p=new Police();  // Console.WriteLine(p.GetSecret());  Console.WriteLine(Police.code);  Console.ReadLine();  }  }  } |
| **Research and write what is the use of sealed class.** |
| * Sealed class exactaly same as normal class and it access variables ,methods,properties etc. * A sealed class cannot be used as base class for other class. |

|  |
| --- |
| **2. Research and write what is the difference between normal properties and auto-implemented**  **properties.** |
| **Normal Properties :** |
| * Normal properties are refer to private variables. |
| **Auto Implemented Properties :** |
| * AutoImplemented Properties are refer to both get and set is required ,some time only (get) will have refered. * It will have not mandatory to access private variables. * For suppose get remove that time showing the error in set . |

|  |
| --- |
| **2(a)WACP to illustrate normal properties ?** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Properties  {  class AvgSpeed  {  private int speed;  private int time;  private int distance;  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)  {  AvgSpeed obj = new AvgSpeed();  obj.Distance = 500;  obj.Time = 50;  Console.WriteLine("Enter the AvgSpeed is :{0}",obj.Speed);  Console.ReadLine();  }  }  } |
| Output :  Screenshot (235) |

|  |
| --- |
| **2(b)WACP to issustrate auto-implemented properties** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Properties  {  class AvgSpeed  {    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)  {  AvgSpeed obj = new AvgSpeed();  obj.Distance = 840;  obj.Time = 20;  Console.WriteLine("Display the Auto-Implement AvgSpeed is :{0}",obj.Speed);  Console.ReadLine();  }  }  } |
| Output :  Screenshot (237) |

|  |
| --- |
| **3. 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 i, n=11;  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 :  Screenshot (239) |

|  |
| --- |
| **4. 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 Day14Project4  {  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 :  Screenshot (241) |

|  |
| --- |
| **5. 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 using\_forloop\_break  {  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 :  Screenshot (244) |