Practical 2: Working with object oriented C# and ASP.NET

Practical 2A

Aim: Create a simple application to perform following operations

1. Function Overloading:

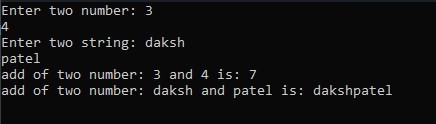
write a c# program which has 2 methods “add” and “con” which can add two number and concatenate two strings which are taken as input from the user and display the output

Code:

|  |
| --- |
| using System;  using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading.Tasks;  namespace Prac2A  { class Overloading  { public void add(int a, int b)  {  Console.WriteLine("add of two number: " + a + " and " + b + " is: " + (a + b)); }  public void add(string str1, string str2)  {  Console.WriteLine("add of two number: " + str1 + " and " + str2 + " is: " + (str1 + str2));  }  }  class Program  { static void Main(string[] args)  {  Console.Write("Enter two number: "); int a = Convert.ToInt32(Console.ReadLine()); int b = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter two string: "); string str1 = Console.ReadLine(); string str2 = Console.ReadLine();  Overloading obj1 = new Overloading(); obj1.add(a, b); obj1.add(str1, str2);  Console.ReadLine();  }  }  } |

Subject: ASP.NET Date: 12/12/23

Output:



1. Constructor overloading

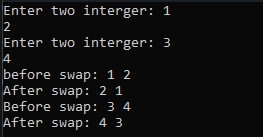
Write a C# program which has 2 constructors which can swap two integers or two floating number which are taken as input from the user and display the output

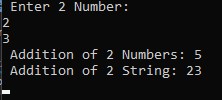
Code:

|  |
| --- |
| using System;  using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading.Tasks;  namespace Prac2A  { class Swap  {  Swap(int n, int m)  {  Console.WriteLine("before swap: " + n + " " + m); int t; t = n; n = m; m = t;  Console.WriteLine("After swap: " + n + " " + m);  }  Swap(double f1, double f2)  {  Console.WriteLine("Before swap: " + f1 + " " + f2); double f; f = f1; f1 = f2; f2 = f;  Console.WriteLine("After swap: " + f1 + " " + f2);  }  static void Main(string[] args)  {  Console.Write("Enter two interger: "); int n = Convert.ToInt32(Console.ReadLine()); int m = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter two interger: "); double f1 = Convert.ToDouble(Console.ReadLine()); double f2 = Convert.ToDouble(Console.ReadLine());  Swap obj1 = new Swap(n, m);  Swap obj2 = new Swap(f1, f2);  Console.Read();  }  }  } |

Subject: ASP.NET Date: 12/12/23

Output:



1. Interfaces Code:

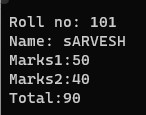
|  |
| --- |
| using System; namespace Prac2A  { interface calculator  { void add(int a, int b);  }  public class DemoNum : calculator  { public void add(int a, int b)  {  Console.WriteLine(" Addition of 2 Numbers: " +(a + b));  }  }  public class DemoString : calculator  { public void add(int a, int b)  {  Console.WriteLine(" Addition of 2 String: " +a + b);  }  } public class interfaces  { public static void Main(string[] arg)  {  Console.WriteLine(" Enter 2 Number: "); int a = Convert.ToInt32(Console.ReadLine()); int b = Convert.ToInt32(Console.ReadLine());  calculator c = new DemoNum(); calculator c1 = new DemoString(); c.add(a, b); c1.add(a, b); Console.ReadLine();  }  }  } |

Output:

1. Inheritance (all types) Code:

|  |
| --- |
| namespace Prac2B  {  class Program  {  static void Main(string[] args)  {  Result r1 = new Result(101, "sARVESH", 50, 40); r1.display();  }  }  } |
| namespace Prac2B  {  internal class Student  {  int roll\_no; string name;  public Student(int roll\_no, string name)  {  this.roll\_no = roll\_no;  this.name = name;  }  public Student() { }  public void display()  {  Console.WriteLine("Roll no: " + roll\_no);  Console.WriteLine("Name: " + name);  }  }  } |
| namespace Prac2B  {  internal class Result:Test  { int total;  public Result(int roll\_no,string name,int marks1,int marks2) |

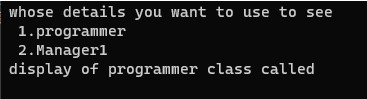
|  |
| --- |
| :base(roll\_no, name, marks1,marks2)  {  total = getMarks1() + getMarks2();  }  public void display()  { base.display();  Console.WriteLine("Total:" + total);  }  }  } |
| namespace Prac2B  {  internal class Test:Student  {  int marks1, marks2;  public Test(int roll\_no,string name, int marks1, int marks2) :base(roll\_no,name)  {  this.marks1 = marks1; this.marks2 = marks2;  }  public int getMarks1()  {  return marks1;  }  public int getMarks2()  {  return marks2;  }  public void display()  { base.display();  Console.WriteLine("Marks1:" + marks1);  Console.WriteLine("Marks2:" + marks2);  }  }  } |



1. Hierarchical Inheritance Code:

|  |
| --- |
| namespace Prac2B  {  internal class Employee  {  public virtual void display()  {  Console.WriteLine("Display of employee class called:");  }  }  } |
| namespace Prac2B  {  class Pragammer:Employee  {  public void display()  {  Console.WriteLine("Display of programmer class called"); Console.ReadLine();  }  }  } |
| namespace Prac2B  {  class Manager:Employee  {  public void display()  {  Console.WriteLine("Display of manager class called:"); Console.ReadLine();  } |

|  |
| --- |
| }  } |
| class Program  {  static void Main(string[] args)  {  Pragammer objProgrammer;  Manager objManager;  Console.WriteLine("Whose details you want to see \n 1.Pragrammer \n 2.Manager"); int choice=int.Parse(Console.ReadLine());  if (choice == 1)  {  objProgrammer = new Pragammer();  objManager = new Manager();  }  else if(choice == 2)  {  objManager = new Manager();  objProgrammer= new Pragammer();  }  else  {  Console.WriteLine("Wrong choice entered"); Console.ReadLine();  }  }  } |



1. Multiple Inheritance Code:

|  |
| --- |
| using System;  using System.Collections.Generic; using System.Linq; using System.Text;  using System.Threading.Tasks; namespace prac2Aiv  {  class Emp  {  string name;  public Emp(string name)  {  this.name = name;  }  public int BasicSal(int basicSal)  {  return basicSal;  }  public void ShowData()  {  Console.WriteLine(" Name: " + name);  }  }  } |
| using System;  using System.Collections.Generic; using System.Linq; using System.Text;  using System.Threading.Tasks; namespace prac2Aiv  {  class Salary : Emp, Gross  {  int hra;  public Salary(string name, int hra) : base(name)  {  this.hra = hra; |

|  |
| --- |
| }  public int ta  {  get { return S\_ta; }  set { S\_ta = value; }  }  private int S\_ta;  public int da  {  get { return S\_da; }  set { S\_da = value; }  }  private int S\_da;  public int GrossSal()  {  int gSal;  gSal = hra + ta + da + BasicSal(15000); return gSal;  }  public void dispSal()  {  base.ShowData();  Console.WriteLine( "Gross Sal:" +GrossSal());  }  }  } |
| using System;  using System.Collections.Generic; using System.Linq; using System.Text;  using System.Threading.Tasks; namespace prac2Aiv  {  interface Gross  {  int ta  { get; set;  } |
| int da  { get; set;  }  int GrossSal();  }  } |
| using Prac2B; using System;  using System.Collections.Generic; using System.Linq; using System.Text;  using System.Threading.Tasks; namespace prac2Aiv  {  class Program  {  static void Main(string[] args)  {  Salary s = new Salary("sARVESH" , 35000); s.da = 20000;  s.ta = 30000;  s.dispSal();  Console.Read();  }  }  } |

Output:



|  |  |
| --- | --- |
| Name : Sarvesh Sawant | Class: SYIT |
| Subject: ASP.NET | Date: 12/12/23 |

|  |  |
| --- | --- |
| Name : Sarvesh Sawant | Class: SYIT |
| Subject: ASP.NET | Date: 12/12/23 |

Practical 2B

Aim: Create a simple application to demonstrate use of following concepts

i) using delegates and events

Code:

|  |
| --- |
| using System;  using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading.Tasks;  namespace practical2Bi  {  public delegate void TrafficDel();  class Traffic  {  public static void Yellow()  {  Console.WriteLine(&quot;Yellow light signals to get ready&quot;);  }  public static void Green()  {  Console.WriteLine(&quot;Green light signals to go&quot;);  }  public static void Red()  {  Console.WriteLine(&quot;Red light signals to stop&quot;);  }  TrafficDel[] td = new TrafficDel[3]; public void IdentifySignal()  {  td[0] = new TrafficDel(Yellow); td[1] = new TrafficDel(Green);  td[2] = new TrafficDel(Red);  }  public void display()  { td[0](); td[1](); td[2]();  }  }  } |
| namespace practical2Bi  { |

class Program

{

static void Main(string[] args)

{

Traffic ts = new Traffic(); ts.IdentifySignal(); ts.display();

Console.Read();

}

|  |
| --- |
| using System;  using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading.Tasks;  namespace practical2Bi  {  class NotEvenException: Exception  {  public NotEvenException(string msg): base(msg)  {  }  }  } |
| using practical2Bi; using System;  using System.Collections.Generic; using System.Linq; using System.Text;  using System.Threading.Tasks;  namespace practical2B  {  class Program |

}

}

Output:



ii)WRITE a program to accpt a number from th usr and throw an exception if the number is not an even number

Code:

|  |
| --- |
| {  static void Main(string[] args)  {  int num;  try {  Console.Write(&quot;Enter a number: &quot;); num = int.Parse(Console.ReadLine());  if ((num % 2) != 0) throw new NotEvenException(&quot;Not an even number &quot;); else  Console.WriteLine(&quot;Its even number &quot;);  }  catch (NotEvenException e) { Console.WriteLine(e.Message); } Console.Read();  }  }  } |

Output:

