|  |
| --- |
| Day9 Morning Assignment  By  Anusha Bellala |

|  |
| --- |
| 1. Write a C# program to read input from user and print  a. factorial of a number  b. factors of a number  c. check if it prime or not |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApp1  {  class MathsOperations  {  private int input;  public void ReadInput()  {  Console.WriteLine("Enter any number:");  input=Convert.ToInt32(Console.ReadLine());  }  public void Factorial()  {  int fact = 1;  for(int i = 1; i <= input; i++)  {  fact= fact \* i;  }  Console.WriteLine("factorial="+fact);  }  public void PrintFactors()  {  for(int i = 1; i <= input; i++)  {  if(input%i== 0)  Console.WriteLine("Factors="+i);  }  }  public bool IsPrime()  {  int count = 0;  for(int i = 1; i <= input;i++)  {  if(input%i==0)  count++;  }  if(count==2)  return true;  else  return false;  }  }  internal class Program  {  static void Main(string[] args)  {  MathsOperations obj = new MathsOperations();  obj.ReadInput();  obj.Factorial();  obj.PrintFactors();  if(obj.IsPrime())  Console.WriteLine("Input is prime number");  else  Console.WriteLine("Input is not a prime number");    Console.ReadLine();  }  }  } |
| Ouput: |

|  |
| --- |
| 2. Write C# program to read two numbers from use and print  a. sum of two numbers  b. difference of two numbers  c. product of two numbers  d. division of two numbers. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApp2  {  class MathTask  {  private int a;  private int b;  public void ReadData()  {  Console.WriteLine("Enter first number:");  a=Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter second number:");  b=Convert.ToInt32(Console.ReadLine());  }  public int AddNumbers()  {  return a + b;  }  public int SubNumbers()  {  return a- b;  }  public int MultiplyNumbers()  {  return a \* b;  }  public int DivisionNumbers()  {  return a / b;  }  }  internal class Program  {  static void Main(string[] args)  {  MathTask obj=new MathTask();  obj.ReadData();  Console.WriteLine(obj.AddNumbers());  Console.WriteLine(obj.SubNumbers());  Console.WriteLine(obj.MultiplyNumbers());  Console.WriteLine(obj.DivisionNumbers());  Console.ReadLine();  }  }  } |
| Ouput: |

|  |
| --- |
| 3. Create an employee class with below variables  id, name, salary, company  write methods to read data and print data. |
| Code: |
| using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApp3  {  class Employee  {  public int id;  public string name;  public int salary;  public static string company = "NationsBenefits";  /// <summary>  /// Read data from user  /// </summary>  public void ReadData()  {  Console.WriteLine("Enter Employee id:");  id = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter Employee name:");  name = Console.ReadLine();  Console.WriteLine("Enter Employee salary:");  salary = Convert.ToInt32(Console.ReadLine());  }  /// <summary>  /// Print data  /// </summary>  public void PrintData()  {  Console.WriteLine($"Id={id},Name={name},Salary={salary},Company={company}");  }  }  internal class Program  {  static void Main(string[] args)  {  Employee emp = new Employee();  emp.ReadData();  emp.PrintData();    Console.ReadLine();  }  }  } |
| Output: |

|  |
| --- |
| 4. Research and find the difference between normal variable and static variable. |

|  |  |
| --- | --- |
| **Static Variable** | **Non-static (normal) Variable** |
| Static variables is also known as class variables. | Normal variables is also known as class variables. |
| Static variables is called using class name. | Normal variables doesn’t use class name for calling. |
| We cannot call static variable with objects. | Normal variables can call objects. |
| Static variables initialize only once of a class. | Normal variables 0 or n no. of times ,depending on the no. of objects created for that class. |
| If you want a variable to have the same value throughout all instances of a class then you need to declare that variable as a static variable with a static keyword. | Normal variables does not declare static keyword. |

|  |
| --- |
| 5. Write 5 points discussed about constructor |

|  |
| --- |
| 1.Constructor has same name as the class itself |
| 2.Constructors don’t have return type |
| 3.A constructor is automatically called when an object is created. |
| 4.A constructor is used to initialize class variable while creating objects. |
| 5.If we do not specify a constructor, C# compiler generates a default constructor for object (expects no parameters and has an empty body). |

|  |
| --- |
| 6. Create Employee class with two constructors as discussed in the class. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApp3  {  class Employee  {  public int id;  public string name;  public int salary;  public static string company = "NationsBenefits";  public Employee()  {  id = 0;  name = null;  salary = 0;  }  public Employee(int id, string name, int salary)  {  this.id = id;  this.name = name;  this.salary = salary;  }  /// <summary>  /// Read data from user  /// </summary>  public void ReadData()  {  Console.WriteLine("Enter Employee id:");  id = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter Employee name:");  name = Console.ReadLine();  Console.WriteLine("Enter Employee salary:");  salary = Convert.ToInt32(Console.ReadLine());  }  /// <summary>  /// Print data  /// </summary>  public void PrintData()  {  Console.WriteLine($"Id={id},Name={name},Salary={salary},Company={company}");  }  }  internal class Program  {  static void Main(string[] args)  {  Employee emp1 = new Employee(1,"siddu",8000);  emp1.PrintData();  Employee emp2 = new Employee();  emp2.PrintData();      Console.ReadLine();  }  }  } |
| Ouput: |