**Tutorial – 3**

1. Draw a real picture for class and object. Differentiate class and object in terms of diagram only.

Perform following tasks.

Task 1: Create a class

Task 2: Add few data members as private

Task 3: Add few methods as public to work on defined data members

Task 4: Create a Demo class with main method.

Task 5: Create at least two objects of a class defined in Task 1 into main method and call all methods using that object.

Task 6: Write comment for each important portion of code like data members’ declaration, methods, some important logic etc.

Task 7: Summarize above solution in your own few words to visualize the solution to the end user.

using System;

namespace Tutorial\_3

{

class Employee

{

private int id = 0;

String name,position, expertise;

public void getdata(int ID)

{

id = ID;

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("Enter ID: {0}",id);

Console.Write("Enter Name: ");

name = Console.ReadLine();

Console.Write("Enter Position: ");

position = Console.ReadLine();

Console.Write("Expertise in which field: ");

expertise = Console.ReadLine();

}

public void printdata()

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("ID: " + id);

Console.WriteLine("Name: " + name);

Console.WriteLine("Position: " + position);

Console.WriteLine("Expertise: " + expertise);

}

}

class Program

{

static void Main(string[] args)

{

Employee e1 = new Employee();

Employee e2 = new Employee();

e1.getdata(1);

e2.getdata(2);

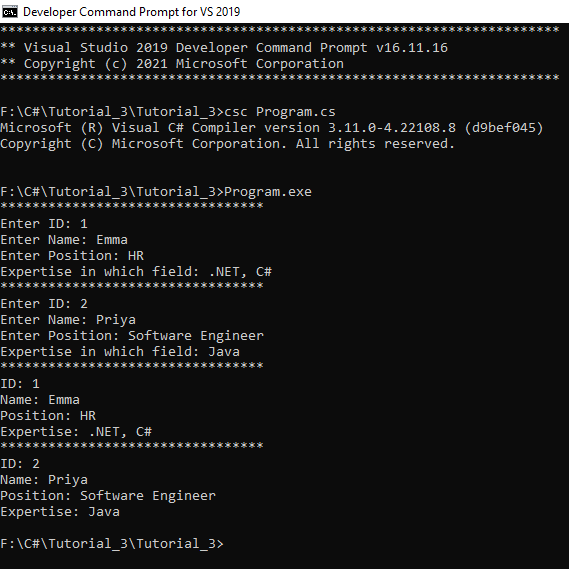
e1.printdata();

e2.printdata();

}

}

}

**Output:**

1. Define a class Clock with three *private* integer data members hour, min and sec. Define a no argument constructor to initialize time value to 12:00:00. Define a three-argument constructor to initialize the time.

Define methods to

* + 1. Increment time to next second.
    2. Display the time value.
    3. Return the hour (*int getHour()*)
    4. Return the minute (*int getMinute()*)
    5. Return the seconds (*int getSeconds()*)

using System;

using System.Collections.Generic;

using System.Text;

namespace Tutorial\_2

{

class Program2

{

static void Main(string[] args)

{

/\* local variable definition \*/

int a = 100;

/\* check the boolean condition \*/

if (a < 20)

{

/\* if condition is true then print the following \*/

Console.WriteLine("a is less than 20");

}

else

{

/\* if condition is false then print the following \*/

//………………………………Missing statement-1……………………………….//

Console.WriteLine("a is not less than 20");

}

//………………………………Missing statement-2……………………………….//

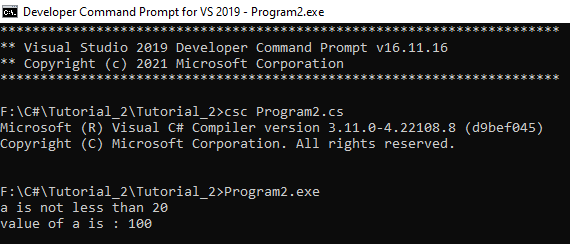
Console.WriteLine("value of a is : {0}",a);

Console.ReadLine();

}

}

}

**Output:**

1. **Define a Student class with appropriate data members, property, constructors, methods etc. Define another class called TestStudent within the same .cs file. Also create an object of student class and demonstrate the use of student class.**

using System;

namespace Tutorial\_2

{

class Program3

{

static void Main(string[] args)

{

string firstName = "John";

string lastName = "Doe";

Console.WriteLine("Name: " + firstName + " " + lastName);

Console.WriteLine("Please enter a new first name:");

firstName = Console.ReadLine();

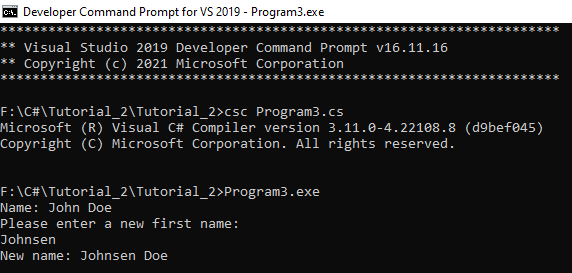
Console.WriteLine("New name: " + firstName + " " + lastName);

Console.ReadLine();

}

}

}

**Output:**

1. **Use above program classes and create objects for 5 students and demonstrate the use student class.**

using System;

using System.Collections.Generic;

using System.Text;

namespace Tutorial\_2

{

class Program4

{

static void Main(string[] args)

{

int x=10, y=3;

Console.WriteLine("\n----->Arithmetic Operators<-----");

Console.WriteLine("--------------------------------");

Console.Write("->>X:{0}", x);

Console.WriteLine(" ->>Y:{0}", y);

Console.WriteLine("Addition of X and Y: {0}" , x + y);

Console.WriteLine("Subtraction of X and Y: {0}", x - y);

Console.WriteLine("Multiplication of X and Y: {0}", x \* y);

Console.WriteLine("Division of X and Y: {0}", x / y);

Console.WriteLine("Modulus of X and Y: {0}", x % y);

Console.WriteLine("Increment of X: {0}", x++);

Console.WriteLine("Decrement of X: {0}", x--);

int a=29;

Console.WriteLine("\n---->Assignment Operators<----");

Console.WriteLine("--------------------------------");

Console.WriteLine("->>A:{0}", a);

Console.WriteLine("A= operator: {0}", a);

Console.WriteLine("A+=3 operator: {0}", a+=3);

Console.WriteLine("A-=3 operator: {0}", a-=3);

Console.WriteLine("A\*=3 operator: {0}", a\*=3);

Console.WriteLine("A/=3 operator: {0}", a /= 3);

Console.WriteLine("A%=3 operator: {0}", a %= 3);

Console.WriteLine("A&=3 operator: {0}", a &= 3);

Console.WriteLine("A|=3 operator: {0}", a |= 3);

Console.WriteLine("A^=3 operator: {0}", a ^= 3);

Console.WriteLine("A>>=3 operator: {0}", a >>= 3);

Console.WriteLine("A<<=3 operator: {0}", a <<= 3);

int x1 = 20, y1 = 32;

Console.WriteLine("\n----->Comparison Operators<-----");

Console.WriteLine("----------------------------------");

Console.Write("->>X:{0}", x1);

Console.WriteLine(" ->>Y:{0}", y1);

Console.WriteLine("Equal to X and Y: {0}", x1 == y1);

Console.WriteLine("Not equal X and Y: {0}", x1 != y1);

Console.WriteLine("Greater than X and Y: {0}", x1 > y1);

Console.WriteLine("Less than X and Y: {0}", x1 < y1);

Console.WriteLine("Greater than or equal to X and Y: {0}", x1 >= y1);

Console.WriteLine("Less than or equal to X and Y: {0}", x1 <= y1);

int a1 = 5;

Console.WriteLine("\n----->Logical Operators<-----");

Console.WriteLine("-------------------------------");

Console.WriteLine("->>A:{0}", a1);

Console.WriteLine("Logical and operator: {0}", a1 < 5 && a1 < 10);

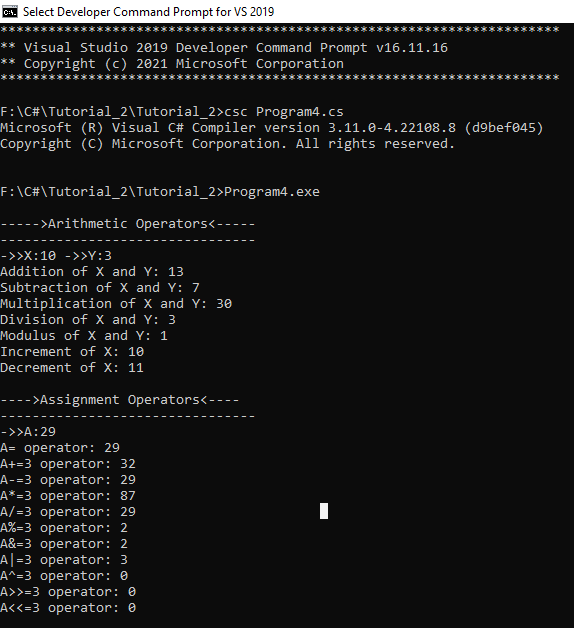
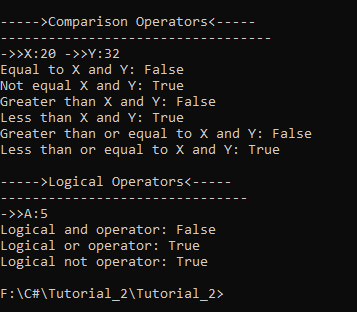
Console.WriteLine("Logical or operator: {0}", a1 < 5 || a1 < 10);

Console.WriteLine("Logical not operator: {0}", !(a1 < 5 && a1 < 10));

}

}

}

**Output:**

1. Rearrange the given code to get the desired output.

using System;

namespace ConsoleApplication

{

    class Product

    {

        public Product(int pcd, String pnm, String mnm)

        {

            mname = mnm;

        }

        public void Display()

        {

            Console.WriteLine("\nManufacturer Name:= " + mname);

        }

    }

  int pcode;

  String pname, mname;

  Console.WriteLine("\nProduct Code:= " + pcode);

  Console.WriteLine("\nProduct Name:= " + pname);

  pcode = pcd;

  pname = pnm;

    public class TestProduct

    {

        public static void Main(string[] args)

        {

            int n = args.Length;

            if (n < 3)

            {

                Console.WriteLine("Syntax Error\n");

                Console.WriteLine("Must Have THREE Arguments\n");

                Console.WriteLine("Please, Write as [csc TestProduct ProductCode ProductName Manufacturer] \n");

            }

            else

            {

                Product p=new Product(pcd,pnm,mnm);

                p.Display();

                Console.Read();

                int pcd = Convert.ToInt32(args[0]);

                String pnm = args[1];

                String mnm = args[2];

            }

        }

    }

}

Output:

Product Code:= P001

Product Name:= Mouse

Manufacturer Name:= Logitech

**Rearrange Code:**

using System;

namespace Tutorial\_3

{

class Product

{

int pcode;

String pname, mname;

public Product(int pcd, String pnm, String mnm)

{

pcode = pcd;

pname = pnm;

mname = mnm;

}

public void Display()

{

Console.WriteLine("\nProduct Code:= " + pcode);

Console.WriteLine("\nProduct Name:= " + pname);

Console.WriteLine("\nManufacturer Name:= " + mname);

}

}

public class TestProduct

{

public static void Main(string[] args)

{

int n = args.Length;

if (n < 3)

{

Console.WriteLine("Syntax Error\n");

Console.WriteLine("Must Have THREE Arguments\n");

Console.WriteLine("Please, Write as [csc TestProduct ProductCode ProductName Manufacturer] \n");

}

else

{

int pcd = Convert.ToInt32(args[0]);

String pnm = args[1];

String mnm = args[2];

Product p = new Product(pcd, pnm, mnm);

p.Display();

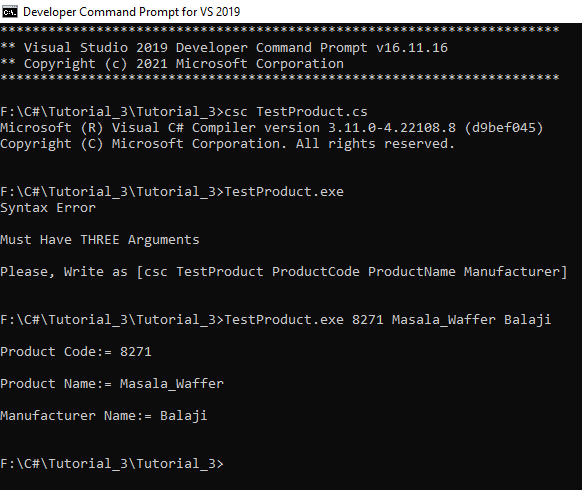
Console.Read();

}

}

}

}

**Output:**

1. Complete the following code that will generate the given output.

using System;

namespace LineApplication

{

   class Line

   {

      private double length;   // Length of a line

      public Line()

      {

          //………………………………Missing statement-1……………………………….//

         //………………………………Missing statement-2……………………………….//

      }

      public void setLength( double len )

      {

          //………………………………Missing statement-3……………………………….//

      }

      public double getLength()

      {

          //………………………………Missing statement-4……………………………….//

      }

}

class TestLine

{

      static void Main(string[] args)

      {

         Line line = new Line();

         // set line length

         Console.WriteLine("Length of line : {0}", line.getLength());

         // set line length

         Console.WriteLine("Length of line : {0}", line.getLength());

         Console.ReadKey();

      }

   }

}

**Updated Code:**

using System;

namespace Tutorial\_3

{

class Line

{

private double length; // Length of a line

public Line()

{

//………………………………Missing statement-1……………………………….//

Console.Write("Object is being created, Length: ");

//………………………………Missing statement-2……………………………….//

length = Convert.ToDouble(Console.ReadLine());

}

public void setLength(double len)

{

//………………………………Missing statement-3……………………………….//

length = len;

}

public double getLength()

{

//………………………………Missing statement-4……………………………….//

return length;

}

}

class TestLine

{

static void Main(string[] args)

{

Line line = new Line();

// set line length

Console.WriteLine("Length of line : {0}", line.getLength());

// set line length

line.setLength(6);

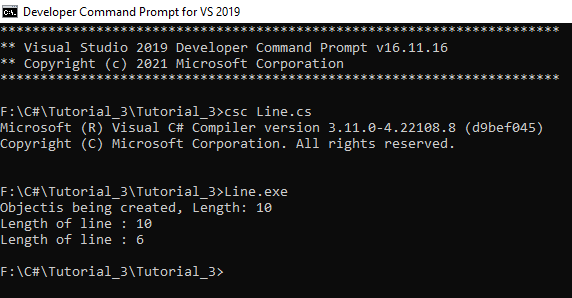
Console.WriteLine("Length of line : {0}", line.getLength());

Console.ReadKey();

}

}

}

**Output:**

1. **Define Enrolment No and Name properties for the Student class and demonstrate use of these properties along with required data members, methods and constructors.**

using System;

namespace Tutorial\_2

{

class Program9

{

static void Main()

{

int[] mat = {1, 2, 3, 5, 4, 1, 2, 3};

int[] a= new int[2];

int m, n;

int count = 0,size= mat.Length;

//int[] mat = new int[size];

//for (int i=0; i<size; i++)

//{

// Console.Write("Enter {0} element: ", i + 1);

// mat[i] = Convert.ToInt32(Console.ReadLine());

//}

Console.Write("array[] = ");

for (int i = 0; i < size; i++)

{

Console.Write("{0} ", mat[i]);

}

for (m = 0; m < size; m++)

{

for (n = 0; n < size; n++)

{

if (mat[m] == mat[n] && m!=n)

{

break;

}

}

if (n == size)

{

a[count] = mat[m];

++count;

}

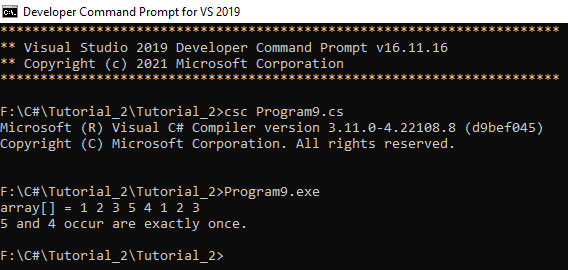
}

Console.WriteLine("\n{0} and {1} occur are exactly once.", a[0], a[1]);

}

}

}

**Output:**