

**Lecture # 3:****Objective:**

- Examples
- Class
- Functions

**Examples:**

Write a program which inputs a string from user and it shows whether string is PALINDROME or not. (Palindrome is a string if read from reverse remain same e.g. madam, radar)

Solution:

```
staticvoid Main(string[] args)
    {

bool flag = false;
Console.WriteLine("Enter String");
string temp = Console.ReadLine();
int j = temp.Length - 1;
for (int i = 0; (i < temp.Length/2); i++)
    {
if(temp[i]!=temp[j-i])
    {
Console.WriteLine("Not a pallindrome");
flag = true;
break;
    }
    }

if(flag==false)
    {
Console.WriteLine("pallindrome");
    }
    }
```

**Classes:**

A class is a construct that enables you to create your own custom types by grouping together variables of other types, methods and events. A class is like a blueprint. It defines the data and behavior of a type.

Classes are declared by using the class keyword, as shown in the following example.

```
public class Customer
{
//Fields, properties, methods and events go here...
}
```

The class keyword is preceded by the access level. Because *public* is used in this case, anyone can create objects from this class. The name of the class follows the class keyword. The remainder of the definition is the class body, where the behavior and data are defined. Fields, properties, methods, and events on a class are collectively referred to as class members.

```
Customer object1 = new Customer();
```

**Constructors in C#**

A class **constructor** is a special member function of a class that is executed whenever we create new objects of that class.

A constructor will have exact same name as the class and it does not have any return type. Following example explains the concept of constructor:

A **default constructor** does not have any parameter but if you need a constructor can have parameters. Such constructors are called **parameterized constructors**. This technique helps you to assign initial value to an object at the time of its creation:

**Example # 1:**

Define a class in C# with following description:

flightNo, destination, distance and fuel (must be private variables). A member function "CALFUEL()" to calculate the value of Fuel as per the following criteria

Distance	Fuel
<=800	500
> 800 and <=1800	1500
> 1800	3000

A function "SETDATA()" allows user to enter values for flight number, destination, distance & call function CALFUEL() to calculate the quantity of fuel. A function "SHOWDATA()" allows user to view the content of all the data members. Write a main function to demonstrate above class functionalities.

## VISUAL PROGRAMMING FALL-2016

```
using System;
using System.Collections.Generic;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Flight airBlue = new Flight();
            airBlue.setData();
            airBlue.showData();
        }
    }

    public class Flight
    {
        private int fNo;
        private string dest;
        private int distance;
        private int fuel;

        public void calFuel()
        {
            if (distance > 1800)
            {
                fuel = 3000;
            }
            elseif (distance > 800)
            {
                fuel = 1500;
            }
            elseif (distance < 800)
            {
                fuel = 500;
            }
        }

        public void setData()
        {
            Console.WriteLine("Enter Flight No.");
            fNo = int.Parse(Console.ReadLine());

            Console.WriteLine("Enter Distance");
            distance = int.Parse(Console.ReadLine());

            Console.WriteLine("Enter Destination");
            dest = (Console.ReadLine());

            calFuel();
        }

        public void showData()
        {
            Console.WriteLine("Flight No is: " + fNo);
            Console.WriteLine("Destination is: " + dest);
        }
    }
}
```

```

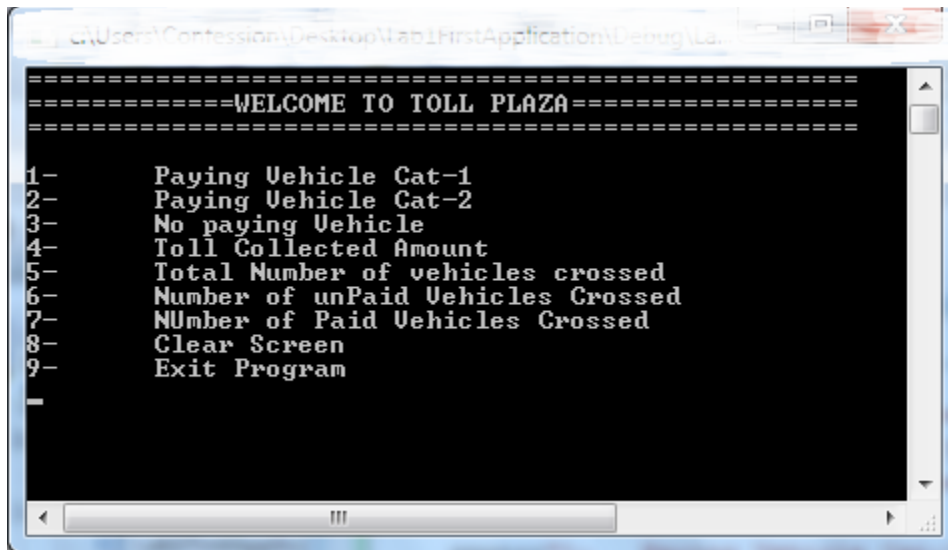
Console.WriteLine("Distance is: " + distance);
Console.WriteLine("Fuel Amount is: " + fuel);
    }

}
}

```

### Practice Example # 1:

Create a menu as shown in figure. There are two categories of paying toll vehicles Cat-1 pays Rs-30 and Cat-2 pays Rs-55. There are no paying vehicles also. Create functions for each of the functionalities given below and show relevant results. This menu continuously runs until user presses 9



### Practice Example # 2:

Create a class called Employee that includes three pieces of information as data members, a name (type string), ID (type integer) and a monthly salary (type integer). Employee class should have a constructor that initializes the three data members. And employee class should have a destructor which shows message when object of the class is destroyed from the memory. Provide a “setData” and a “getData” function for each data member. If the monthly salary is not positive, set it to -1. Write a test program that demonstrates class Employee’s functionalities. Create two Employee objects and display each object’s yearly salary after deducting 4% tax.

### FUNCTIONS:

Block of code in specific format which may reuse again and again in project when required just writing once.

A function is defined with these characteristics:

- Access level - public, private plus some others
- Return value - void or any type such as int
- Method Name - MyGrade
- Any method parameters defined in the brackets () after the method name

```
<visibility><return type><name>(<parameters>)
{<function code>}
```

Let's define a function MyGrade ().

```
public string MyGrade()
{
    return "Grade A";
}
```

### Passing Parameters:

In C#, arguments can be passed to parameters either by value or by reference. Passing by reference enables function members, methods to change the value of the parameters and have that change persist in the calling environment. To pass a parameter by reference, use the ref keyword.

```
class Program
{
    static void Main(string[] args)
    {
        int arg;

        // Passing by value.
        // The value of arg in Main is not changed.
        arg = 4;
        squareVal(arg);
        Console.WriteLine(arg);
        // Output: 4

        // Passing by reference.
        // The value of arg in Main is changed.
        arg = 4;
        squareRef(ref arg);
        Console.WriteLine(arg);
        // Output: 16
    }

    static void squareVal(int valParameter)
    {
        valParameter *= valParameter;
    }

    // Passing by reference
```

```
static void squareRef(ref int refParameter)
{
    refParameter *= refParameter;
}
}
```

**Function overloading:**

Function having same name but different parameters with in same class is known as function overloading.

```
public static int Compare(string strA, string strB);
```

```
public static int Compare(string strA, string strB, bool ignoreCase);
```