**PRACTICAL # 05**

**OBJECT:**

Working with Classes

**THEORY:**

C# is Object-Oriented Programming language that supports Encapsulation, Inheritance and Polymorphism. Central to these features is the concept of classes.

A class is a template or a blueprint form of an object. It specifies both the data and the code that will operate on that data. C# uses a class specification to construct objects. Objects are instances of a class. A class is a logical abstraction. After an object of that class is created, a physical representation of that class is made in memory.

**Class Syntax:**

*class classname {*

*// declare instance variables*

*access type var1;*

*access type var2;*

*access type varN;*

*// ...*

*// declare methods*

*access ret-type method1(parameters) {*

*// body of method*

*}*

*access ret-type method2(parameters) {*

*// body of method*

*}*

*// ...*

*access ret-type methodN(parameters) {*

*// body of method*

*}*

*}*

**Program:**

The program below is a class demo for a building object.

*class Building {*

*public int Floors; // number of floors*

*public int Area; // total square footage of building*

*public int Occupants; // number of occupants*

*}*

An object of this class can be created with new keyword and assigning it to a reference of the class. For example:

*Building house1 = new Building();*

*Building house2 = house1;*

We can access the public members of a class using . operator:

*house.Floors = 2;*

**Class example:**

*class ChkNum {*

*// Return true if x is prime.*

*public bool IsPrime(int x) {*

*if(x <= 1) return false;*

*for(int i=2; i <= x/i; i++)*

*if((x %i) == 0) return false;*

*return true;*

*}*

*// Return the least common factor.*

*public int LeastComFactor(int a, int b) {*

*int max;*

*if(IsPrime(a) || IsPrime(b)) return 1;*

*max = a < b ? a : b;*

*for(int i=2; i <= max/2; i++)*

*if(((a%i) == 0) && ((b%i) == 0)) return i;*

*return 1;*

*}*

*}*

*static void Main() {*

*ChkNum ob = new ChkNum();*

*int a, b;*

*for(int i=2; i < 10; i++)*

*if(ob.IsPrime(i)) Console.WriteLine(i + " is prime.");*

*else Console.WriteLine(i + " is not prime.");*

*a = 7;*

*b = 8;*

*Console.WriteLine("Least common factor for " +a + " and " + b + " is " +*

*ob.LeastComFactor(a, b));*

*a = 100;*

*b = 8;*

*Console.WriteLine("Least common factor for " +a + " and " + b + " is " +*

*ob.LeastComFactor(a, b));*

*a = 100;*

*b = 75;*

*Console.WriteLine("Least common factor for " +a + " and " + b + " is " + ob.LeastComFactor(a, b));*

*}*

**ACTIVITIES**

**Activity 1**

Define a static class called Arithmetic that defines four functions- add, subtract, multiply and divide, each taking in two arguments.

**Activity 2**

Define a class called circle with a member variables called radius, and two member methods called getRadius and setRadius that get and set radius value respectively.

**REVIEW QUESTIONS**

1. Why are classes important in object oriented programming?
2. What are different types of class members possible?
3. What are access modifiers in a class?
4. How do we create an object of a class?
5. When is a class actually loaded in the memory?
6. How can we determine the size of a class object?