



**Karachi Institute of Economics and Technology  
College of Engineering**

(SoftwareEngineering)

**SE3301–SoftwareDesignandArchitecture**

Semester: \_\_\_\_\_

Date of Experiment: \_\_\_\_\_

Student name: \_\_\_\_\_

Faculty Signature: \_\_\_\_\_

Student ID: \_\_\_\_\_

Remarks/Comments: \_\_\_\_\_

Lab12	BehavioralDesignPattern-1:TemplateMethodPattern				
PLOs	PLO1–EngineeringKnowledge	Bloom'sTaxonomy	C1–Recall		
	PLO3–DesignandDevelopment		C3–Apply		
	PLO8–Ethics		P2–Set		
LABTASK PERFORMANCE					
CLO's	AspectsofAssessments	Excellent (75-100%)	Average (50-75%)	Poor (<50%)	Marks
CLO1 10%	<u>Recall</u> Recallthe conceptsofC# Programming Language,OOPandDesignPatternsfromtheory.	Completeunderstandingoftheconcepts of C# ProgrammingLanguage,OOPAndDesign Patterns / actively participatesduringlecture.	Complete understanding of theconcepts of C# ProgrammingLanguage,OOPAndDesign Patterns / less activelyparticipate during lecture.	Studentlacksclearunderstandingo f conceptsofC# ProgrammingLanguage,OOPand Design Patterns / Unable to read andinterpretit.	
CLO5 80%	<u>Engineering Knowledge</u> Demonstrate and illustrateconceptsofBehavioral patterndandexperimentswithtobuild Singleton Patternusing VisualStudio.	Accurately implementthe.net.netto createBehavioral .netto create Behavioral designpattern (TemplateMethod)usingVisualStudioto obtain the correct output as perrequirement/ given tasks.	Implementthe.net.netto createBehavioral designpattern(TemplateMethod)using VisualStudioandgetminorerrorsthat will lead to a slightly differentoutput as per given in a task.	Notabletoimplementthe.net .net to create Behavioral designpattern(TemplateMethod)usingVisual Studio and don'tunderstandhowrequiredoutput andtaskis achieved.	
CLO7 10%	<u>LabSafety</u> Properlyhandle labinfrastructure/safety precautions	Properlyhandlelabequipment& obey safety measures.	Moderatelevellabhandlingand safety measurements	Minorornosafetymeasurementshas been considered.	
TotalMarks:10					

## **Objective:**

- 1) Make attendee understand the basic concept of Behavioral Design Pattern
- 2) Make attendee understand the concept of Template Method Design Pattern
- 3) Make attendee implement example of .net to create Behavioral design pattern (Template Method Pattern)

## **Tools to be used:**

Visual Studio is used to implement .net to create behavioral design pattern (Template Method Pattern).

## **What is Behavioral Design Pattern?**

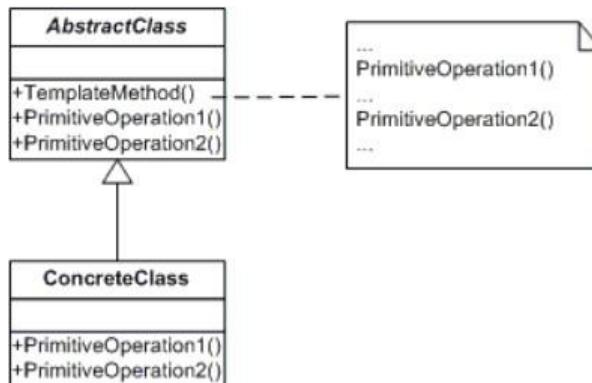
Behavioral Design patterns are the patterns for .Net in which there is a way through which we can pass the request between the chain of objects, or we can say that it defines the manner to communicate between classes and object. Behavioral patterns are those patterns that are concerned with the interaction between the objects. The interaction between the objects should be in such a way that they are talking to each other and still are loosely coupled. The loose coupling is the key to n-tier architecture. Following are the types of

1. Template Method Pattern
2. Iterator Pattern
3. Observer Pattern
4. Chain of Responsibility Pattern
5. Command Pattern

## **Template Method Pattern**

Template method pattern defines the skeleton of an algorithm in an operation, deferring some steps to sub-classes. The Template Method lets sub-classes redefine certain steps of an algorithm without changing the algorithm's structure. It allows you to define a skeleton of an algorithm in a base class and let subclasses override the steps without changing the overall algorithm's structure. Template Method can be recognized by behavioral methods that already have a "default" behavior defined by the base class.

The UML class diagram for the implementation of the Template method design pattern is given below:



*Template Method Pattern*

There are two components participating in the above pattern diagram

1. Abstract class
2. Concrete class

### 1. Abstract Class (DataObject)

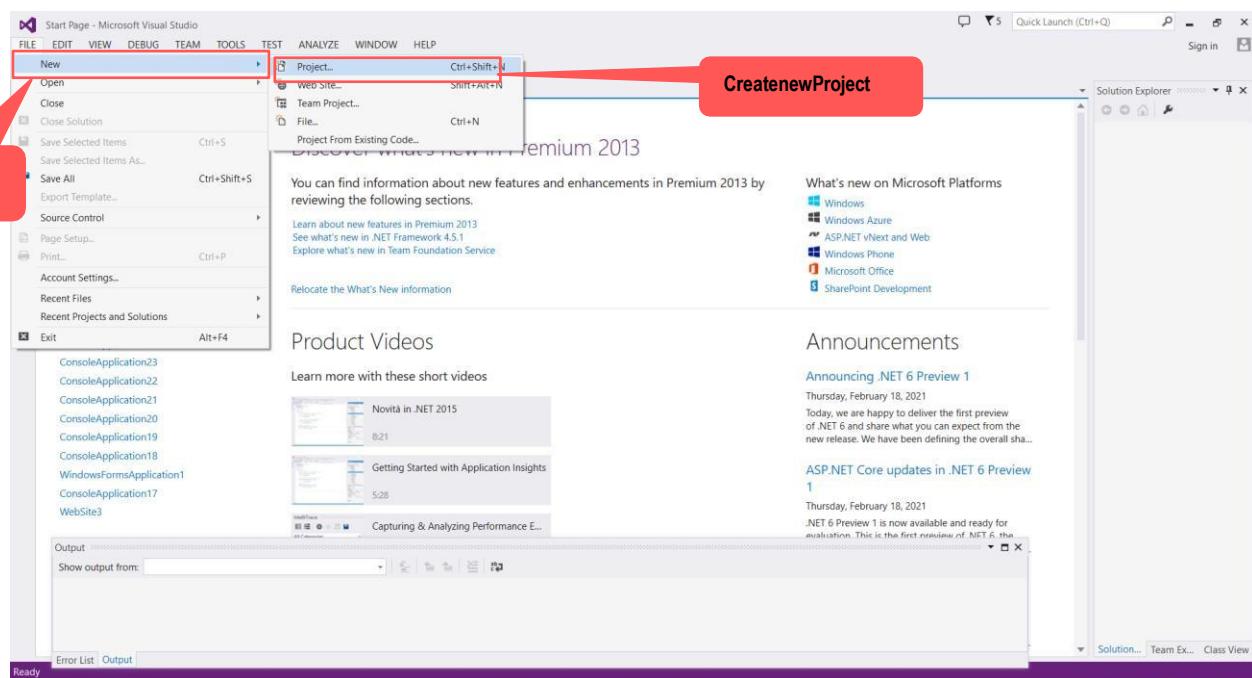
- Defines abstract primitive operations that concrete subclasses define to implement steps of an algorithm
- Implements a template method defining the skeleton of an algorithm. The template method calls primitive operations as well as operations defined in AbstractClass or those of other objects.

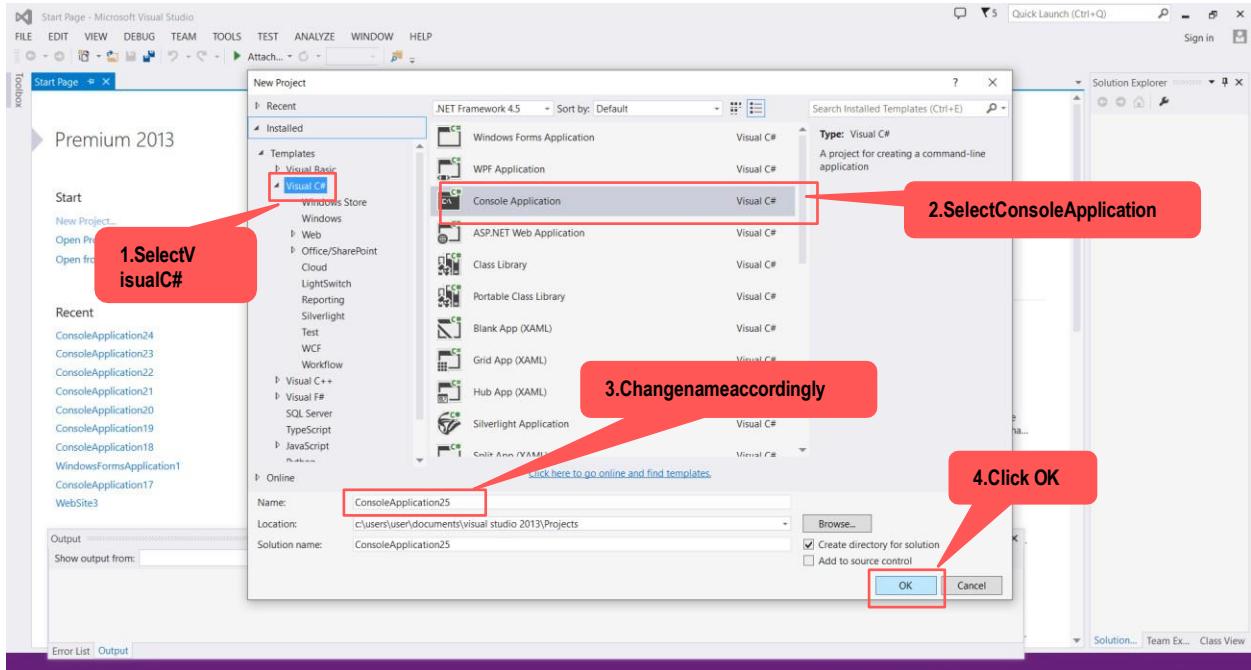
### 2. Concrete Class (CustomerDataObject)

- Implements the primitive operations to carry out subclass-specific steps of the algorithm.

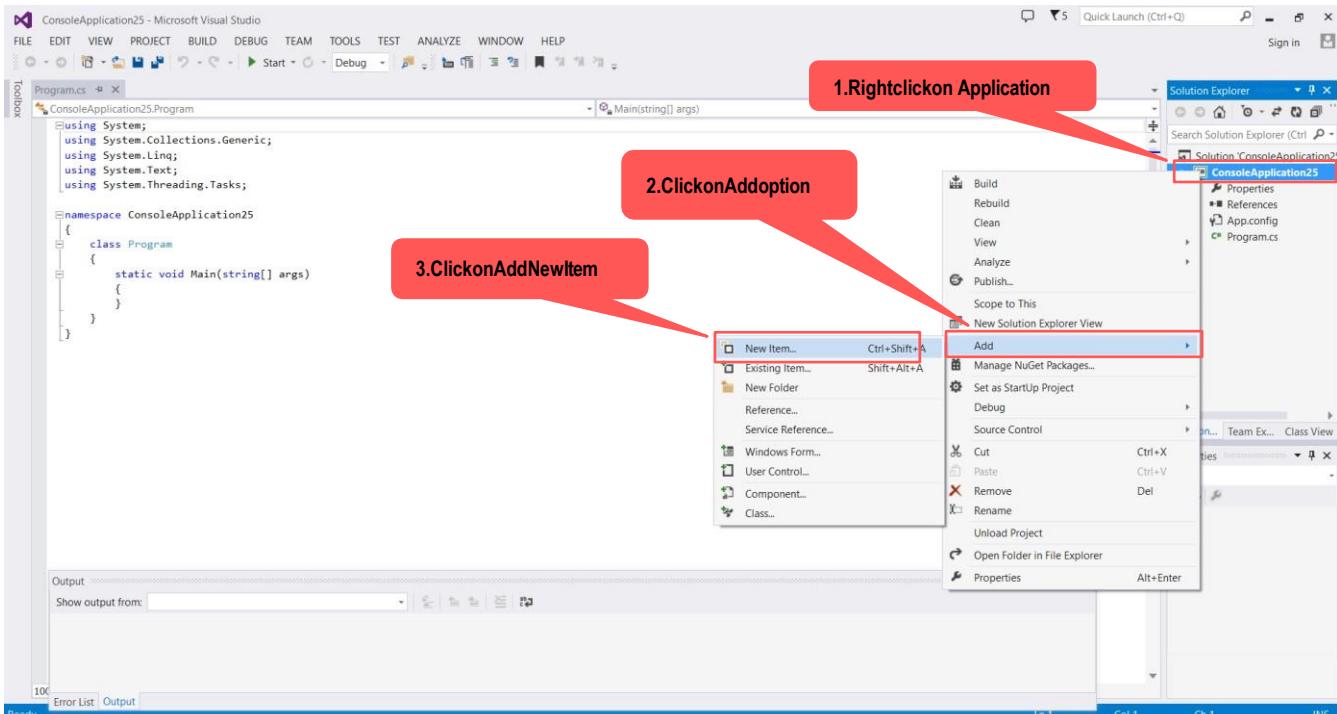
To implement Template Method pattern in real world example, let's follow the following steps

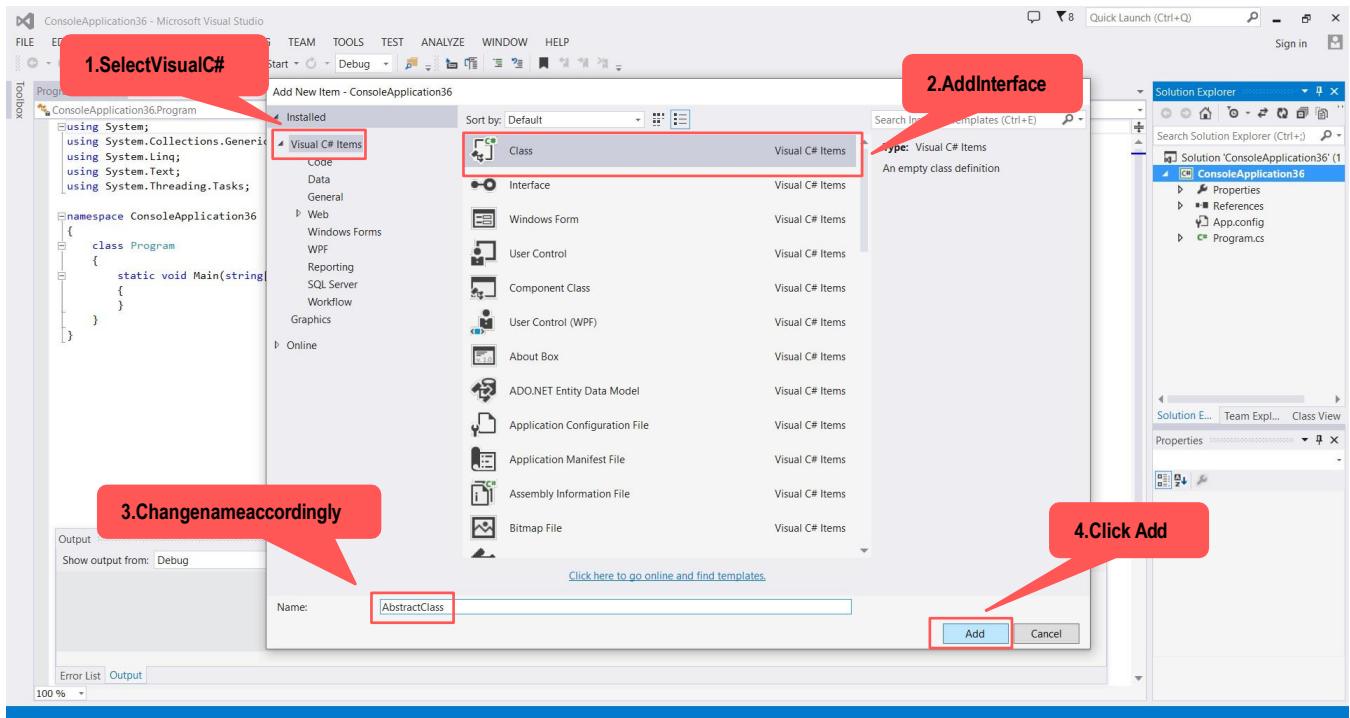
**Step#01:** Create new project of console application.



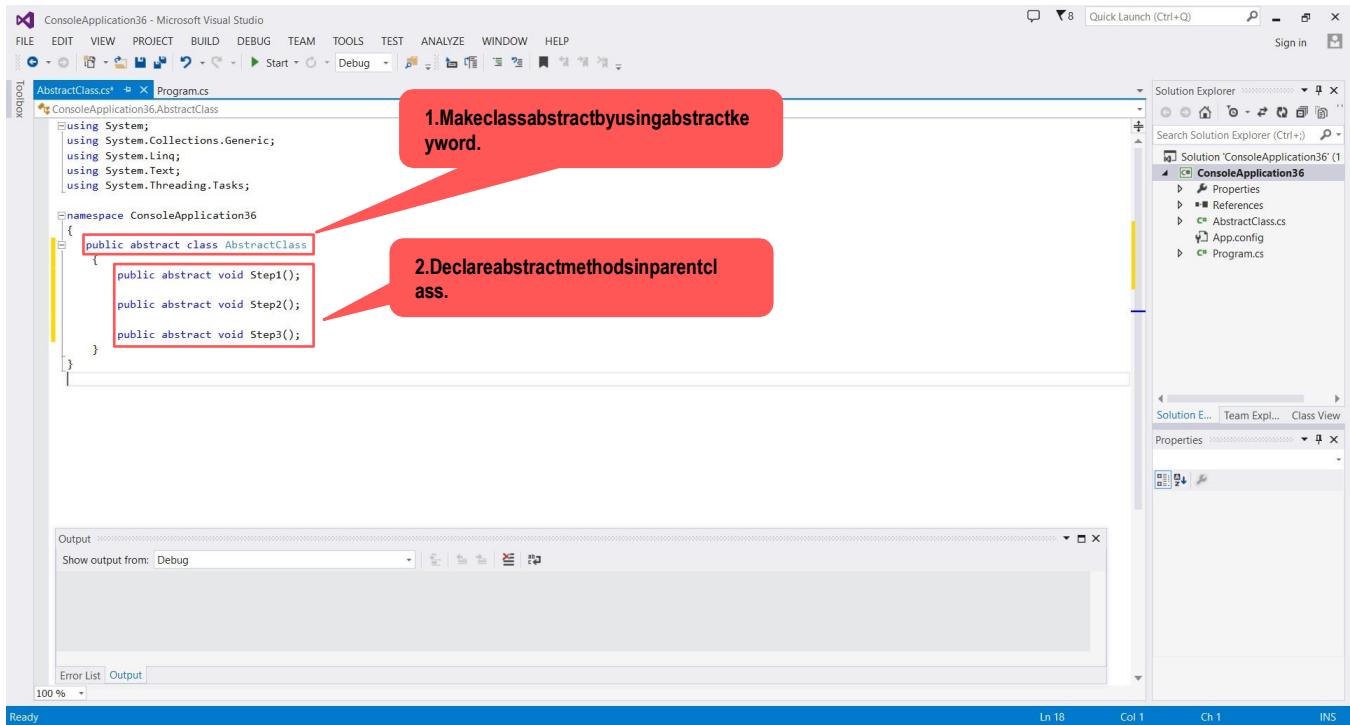


**Step#02:** After creation of Project. Add new class inside the solution. Inside class we will define abstract methods which will override by child classes.

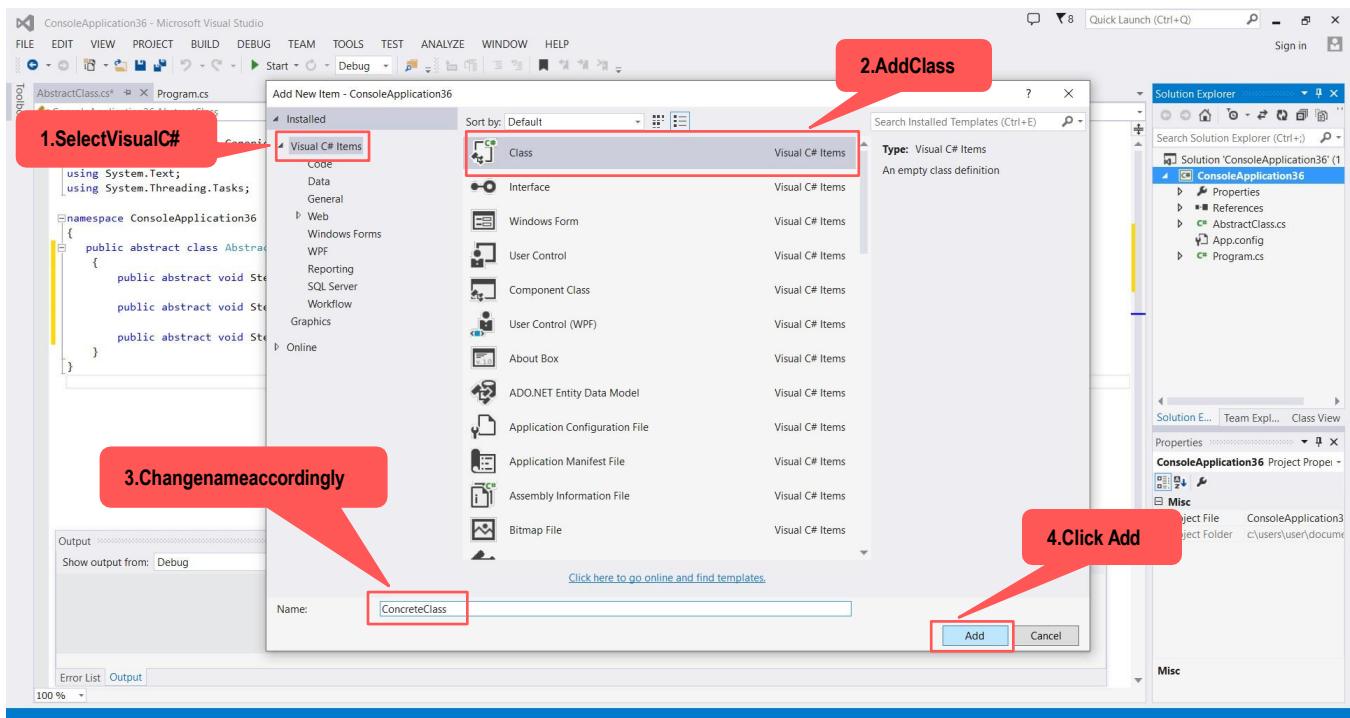
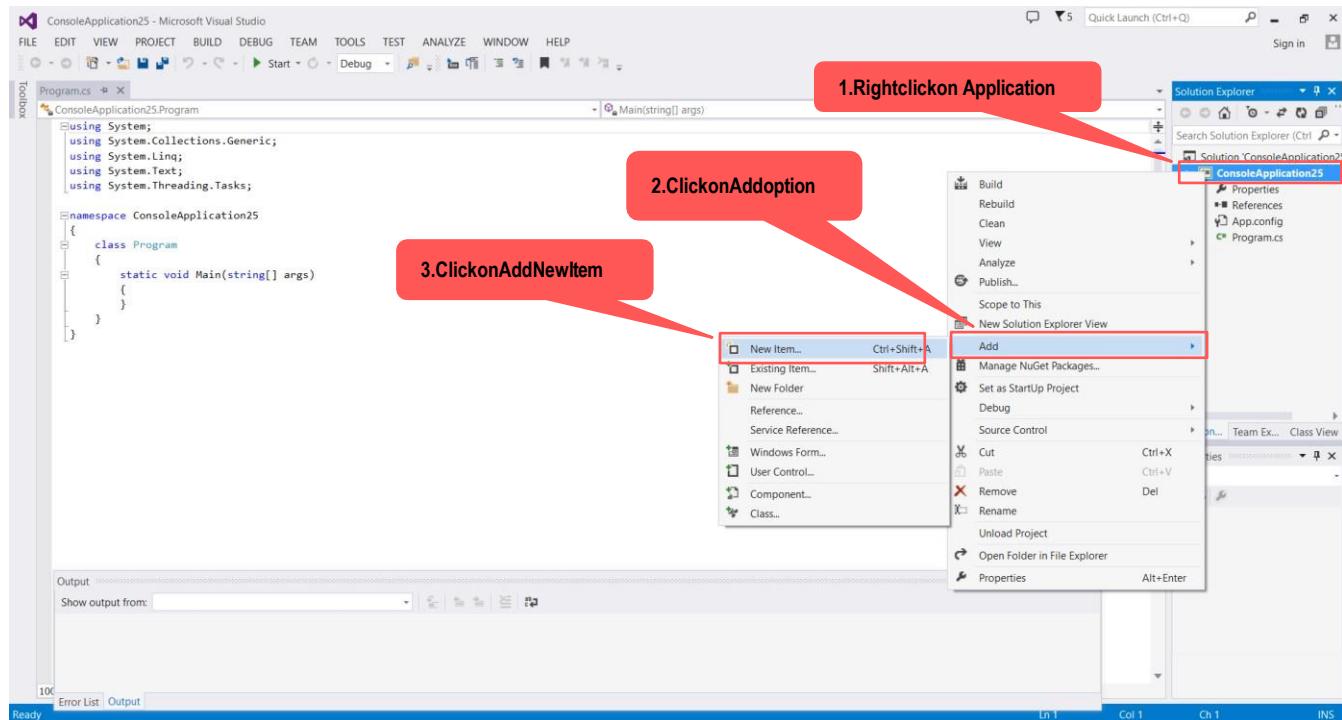




### Step#03:InsideAbstractclass,declareabstractmethodswhichwilloverridebychildclasses.



**Step#04:** Now add Concrete class. The concrete class will contain all the child classes which will inherit the data of parent class.



## Step#05: After adding concrete class, define child classes one by one inside concrete class.

```

ConsoleApplication36 - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
Solution Explorer
Search Solution Explorer (Ctrl+F) ...
Solution 'ConsoleApplication36' (1)
  ConsoleApplication36
    Properties
    References
    AbstractClass.cs
    App.config
    ConcreteClass.cs
    Program.cs
Solution E... Team Expl... Class View
Properties
Ready
100% Error List Output
100% Output
Ln 42 Col 10 Ch 10 INS

```

**1. Child class A inheriting and overriding the methods of parent class**

**2. Child class B inheriting and overriding the methods of parent class**

```

ConsoleApplication36 - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
Solution Explorer
Search Solution Explorer (Ctrl+F) ...
Solution 'ConsoleApplication36' (1)
  ConsoleApplication36
    Properties
    References
    AbstractClass.cs
    App.config
    ConcreteClass.cs
    Program.cs
Solution E... Team Expl... Class View
Properties
Ready
100% Error List Output
100% Output
Ln 42 Col 10 Ch 10 INS

```

## Step#06: Now in main program, create the objects of child classes using parent class and call the methods in order to display the output.

```

ConsoleApplication36 (Running) - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
Process: [2828] ConsoleApplication36.vshost -> [12168] Main Thread
Stack Frame: ConsoleApplication36.RealSubject.Subtract()
ConcreteClass.cs AbstractClass.cs Program.cs
Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication36
{
    class Program
    {
        static void Main(string[] args)
        {
            AbstractClass obj1 = new ConcreteClassA();
            obj1.Step1();
            obj1.Step2();
            obj1.Step3();

            AbstractClass obj2 = new ConcreteClassB();
            obj2.Step1();
            obj2.Step2();
            obj2.Step3();

            Console.ReadKey();
        }
    }
}

```

**1. Create an object of first child class**

**2. calling methods of first child class**

**3. Create an object for second child class**

**4. calling methods of second child class**

**Output of first child class**

**Output of second child class**

```

ConsoleApplication36 - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
Solution Explorer
Search Solution Explorer (Ctrl+F) ...
Solution 'ConsoleApplication36' (1)
  ConsoleApplication36
    Properties
    References
    AbstractClass.cs
    App.config
    ConcreteClass.cs
    Program.cs
Solution E... Team Expl... Class View
Properties
Ready
100% Error List
  0 Errors | 0 Warnings | 0 Messages
  Description
Call Stack Breakpoints Command Window Immediate Window Output Error List Autos Locals Watch 1
File Line Column Project
Ln 29 Col 1 Ch 1 INS

```

**LabTask:**

- 1) Implement Template Method Design Pattern using simple methods (Marks:1)
  - 2) Implement Template Method Design Pattern for following: (Marks:3)
    - Make a parent class and define the method that give square, cube, table and factorial of given number
    - Make three child classes and override the methods given in parent class.
- 

**Home Task:**

- 1) Implement Template Method Design Pattern in University Scenario. (Marks:4)

A university provides discount in fee based upon the student's cgpa. University offers 50% discount upon 4 cgpa, 30% on 3.5 and 15% on 3 cgpa respectively. Create the methods in parent class and make at least three child classes (three departments). Each department will inherit the methods from the parent class. Also display the result when  $\text{cgpa} < 3$ .

Create a program to solve above problem.