**Tradional Approach**

using System;

using System.Collections.Generic;

using System.Text;

namespace CreationalDesignPatternDemos

{

class Teacher

{

public string getName()

{

return "Teacher Name";

}

public string getFName()

{

return "Teacher Father Name";

}

}

class Student

{

public string getName()

{

return "Student Name";

}

public string getFName()

{

return "Student Father Name";

}

}

}

In Main

Teacher teacher = new Teacher();

teacher.getName();

Using Factory

First , you add an Interface

interface i1

{

string getName();

string getfName();

}

Second, you create classes which implements that interface

class Teacher : i1

{

public string getName()

{

return "Teacher Name";

}

public string getFName()

{

return "Teacher Father Name";

}

}

class Student : i1

{

public string getName()

{

return "Student Name";

}

public string getFName()

{

return "Student Father Name";

}

}

-------------------------------------

using System;

using System.Collections.Generic;

using System.Text;

namespace CreationalDesignPatternDemos

{

class Teacher : i1

{

public string getName()

{

return "Teacher Name";

}

public string getFName()

{

return "Teacher Father Name";

}

}

class Student : i1

{

public string getName()

{

return "Student Name";

}

public string getFName()

{

return "Student Father Name";

}

}

interface i1

{

string getName();

string getfName();

}

}

**Now create a Factory Pattern class**

class CreateObj

{

public static i1 getObject(string typeOfObject)

{

i1 obj = null;

if (typeOfObject.ToLower() == "student")

{

obj = new Student();

}

else

{

obj = new Teacher();

}

return obj;

}

}

------------------------------------

using System;

using System.Collections.Generic;

using System.Text;

namespace CreationalDesignPatternDemos

{

class Teacher : i1

{

public string getName()

{

return "Teacher Name";

}

public string getFName()

{

return "Teacher Father Name";

}

}

class Student : i1

{

public string getName()

{

return "Student Name";

}

public string getFName()

{

return "Student Father Name";

}

}

interface i1

{

string getName();

string getFName();

}

class CreateObj

{

public static i1 getObject(string typeOfObject)

{

i1 obj = null;

if (typeOfObject.ToLower() == "student")

{

obj = new Student();

}

else

{

obj = new Teacher();

}

return obj;

}

}

}

-----------------------------------------------------------

Create objects in Main Method

using System;

namespace CreationalDesignPatternDemos

{

class Program

{

static void Main(string[] args)

{

//Tea cher teacher = new Teacher();

//teacher.getFName();

Console.WriteLine("ENter your class type");

string type = Console.ReadLine();

i1 obj = CreateObj.getObject(type);

Console.WriteLine(obj.getName());

}

}

}