**Task No. 1:** Implement Factory pattern for creating pages and popups(Modals, Alerts, etc.) in website for pages separate interface defined and for popups separate interface defined.

**Solution:**

**Main Method:**

Console.WriteLine("=========Select Desired Operation===========");

Console.WriteLine("What do you want to look:\n1)Page \n2)Pop-up ");

int x = int.Parse(Console.ReadLine());

Console.WriteLine("");

Console.WriteLine("=======Enter that Page or Pop-up Name========");

string type = Console.ReadLine().ToLower();

switch (x)

{

case 1:

page\_interface pgobj = Program.get\_pageObj(type);

Console.WriteLine(pgobj.get\_welcome());

break;

case 2:

popup\_Interface popobj = Program.get\_popObject(type);

Console.WriteLine(popobj.Be\_carefull());

break;

}

**Interface Page**

interface Page\_interface

{

string get\_welcome();

}

class Home : Page\_interface

{

public string get\_welcome()

{

return "Welcome to Home Page";

}

}

class About : Page\_interface

{

public string get\_welcome()

{

return "Welcome to About Page";

}

}

class Contact : Page\_interface

{

public string get\_welcome()

{

return "Welcome to Contact Page";

}

}

**Interface Pop-up**

interface popup\_Interface

{

string Be\_carefull();

}

class Information : popup\_Interface

{

public string Be\_carefull()

{

return "Welcome to Information Page";

}

}

class Warning : popup\_Interface

{

public string Be\_carefull()

{

return "Welcome to Warning Page \n Be Alert!";

}

}

**Class Program**

public static Page\_interface get\_pageObj(string type)

{

Page\_interface pageObj = null;

if (type.ToLower() == "home")

{

pageObj = new Home();

}

else if (type.ToLower() == "about")

{

pageObj = new About();

}

else if (type.ToLower() == "contact")

{

pageObj = new Contact();

}

else

return null;

return pageObj;

}

public static popup\_Interface get\_popObject(string type)

{

popup\_Interface popObj = null;

if (type.ToLower() == "information")

{

popObj = new Information();

}

else if (type.ToLower() == "warning")

{

popObj = new Warning();

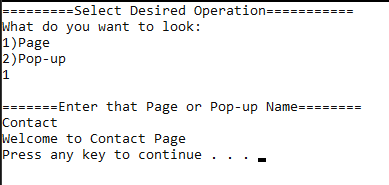
}

else

return null;

return popObj;

**Output:**



**Task No. 2:** Implement factory pattern for creating Ranks/Positions of employees.

**Solution:**

**Interface IEmpoyee**

interface IEmpoyee

{

string getrank();

void details();

}

**Class Employee1**

class Employee1 : IEmpoyee

{

public string getrank()

{

Console.Write("Rank / Postion : ");

return "manager";

}

public void details()

{

Console.WriteLine("Name: Mutayyab");

Console.WriteLine("Hired date: 28-03-2020");

Console.WriteLine("Company Name: Textile Company");

}

}

**Class Employee2**

class Employee2 : IEmpoyee

{

public string getrank()

{

Console.Write("Rank / Postion : ");

return "Developer";

}

public void details()

{

Console.WriteLine("Name: abc");

Console.WriteLine("Hired date: 01-01-2000");

Console.WriteLine("Company Name: Textile Company");

}

}

**Class Object Creation**

class ObjectCreation

{

public static IEmpoyee getObject(string type)

{

IEmpoyee obj = null;

if (type.ToLower() == "employee1")

{

obj = new Employee1();

}

else

{

obj = new Employee2();

}

return obj;

}

}

**Main Method**

class Program

{

static void Main(string[] args)

{

Console.Write("Enter Employee id : ");

string type = Console.ReadLine();

IEmpoyee obj = ObjectCreation.getObject(type);

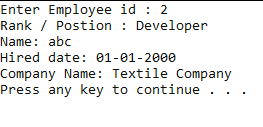
Console.WriteLine(obj.getrank());

obj.details();

}

}

**Output:**

A picture containing text, font, screenshot, white

Description automatically generated