Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

**05**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **01** | **Implement Singleton pattern on printer functionality** |
| **02** | **Implement Singleton pattern for Sessions.** |
| 03 | Implement Singleton pattern for logger application |

Submitted On:

**21 – 4 - 2023**

(Date: DD/MM/YY)

**Task No. 1:** Implement Singleton pattern on printer functionality.

**Solution:**

**Class Printer**

using System;

namespace SDALAB05.CreationalPattern

{

class Printer

{

public static int count = 0;

public static Printer \_obj;

private Printer()

{

}

public static Printer Getobject()

{

if (\_obj == null)

{

\_obj = new Printer();

count++;

}

return \_obj;

}

public void Print()

{

Console.WriteLine("Object Created Of Printer "+count+" Time");

}

}

}

**Main Method**

using SDALAB05.CreationalPattern;

using System;

namespace SDALAB05

{

class Program

{

static void Main(string[] args)

{

Printer PrinterObject = Printer.Getobject();

PrinterObject.Print();

Printer PrinterObject2 = Printer.Getobject();

PrinterObject2.Print();

Printer PrinterObject3 = Printer.Getobject();

PrinterObject3.Print();

Printer PrinterObject4 = Printer.Getobject();

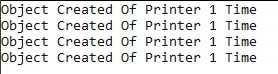
PrinterObject4.Print();

Console.ReadLine();

}

}

}

**Output:**

**Task No. 2:** Implement Singleton pattern for Sessions.

**Solution:**

**Class session**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SDALAB05.CreationalPattern

{

class session

{

public static int count = 0;

public static session \_obj;

private session()

{

}

public static session Getobject()

{

if (\_obj == null)

{

\_obj = new session();

count++;

}

return \_obj;

}

public void Print()

{

Console.WriteLine("Object Created Of session " + count+" Time");

}

}

}

**Main Method**

using SDALAB05.CreationalPattern;

using System;

namespace SDALAB05

{

class Program

{

static void Main(string[] args)

{

session sessionObject1 = session.Getobject();

sessionObject1.Print();

Console.WriteLine("Session 1\n--------Welcome to Facebook---------\n\n");

session sessionObject2 = session.Getobject();

sessionObject2.Print();

Console.WriteLine("Session 2\n-------Please Login here----------\n\n");

session sessionObject3 = session.Getobject();

sessionObject3.Print();

Console.WriteLine("Session 3\n-----------Home---------------\n\n");

session sessionObject4 = session.Getobject();

sessionObject4.Print();

Console.WriteLine("Session 4\n----------Profile---------------\n\n");

session sessionObject5 = session.Getobject();

sessionObject5.Print();

Console.WriteLine("Session 5\n----------Setting---------------\n\n");

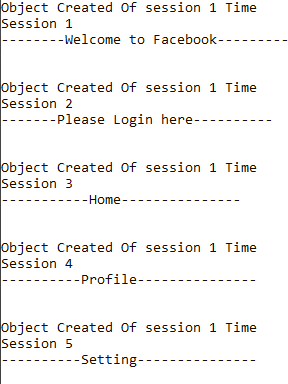
Console.ReadLine();

}

}

}

**Output:**



**Task No. 3:** Implement Singleton pattern for logger application.

**Solution:**

**Class Printer**

using System;

namespace SDALAB05.CreationalPattern

{

class loggger

{

public static int count = 0;

public static loggger \_obj;

private loggger()

{

}

public static loggger Getobject()

{

if (\_obj == null)

{

\_obj = new loggger();

count++;

}

return \_obj;

}

public void Print()

{

Console.WriteLine("Object Created Of Logger " + count+" Time");

}

}

}

**Main Method**

using SDALAB05.CreationalPattern;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SDALAB05

{

class Program

{

static void Main(string[] args)

{

loggger LoggerObject1 = loggger.Getobject();

LoggerObject1.Print();

Console.WriteLine( "Application 1\n========Runs efficiently========\n\n");

loggger LoggerObject2 = loggger.Getobject();

LoggerObject2.Print();

Console.WriteLine("Application 2\n======Found error during the login======\n\n");

loggger LoggerObject3 = loggger.Getobject();

LoggerObject3.Print();

Console.WriteLine("Application 3\n==========Intializing==========\n\n");

loggger LoggerObject4 = loggger.Getobject();

LoggerObject4.Print();

Console.WriteLine("Application 4\n=========Giving Warnings=========\n\n");

loggger LoggerObject5 = loggger.Getobject();

LoggerObject5.Print();

Console.WriteLine("Application 5\n=========Processing Completed=========\n\n");

Console.ReadLine();

}

}

}

**Output:**

