

1908692

by Sai Jangili

FILE	18928_SAI_JANGILI_1908692_226878_1501939907.DOCX (1.63M)		
TIME SUBMITTED	17-FEB-2020 11:10PM (UTC+0000)	WORD COUNT	790
SUBMISSION ID	120256916	CHARACTER COUNT	4792

Advanced Software Development
[ACCA5011]

Name: - Sai Harsha Jangili

Student ID: - 1908692

Table of Contents

The Difference between a Class and an Object	3
Class: -	3
Object: -	3
Difference between Class and an Object: -	3
Data Encapsulation	4
Interfaces	5
Impact of interface in developing a code	5
References	6

The Difference between a Class and an Object

Class: -

A class is a prototype or template of an object which contain variables which stores the required data and function to perform operations, classes are helpful in forming and handling new object. Class is a group that contains similar objects

(Anon., n.d.).



Object: -

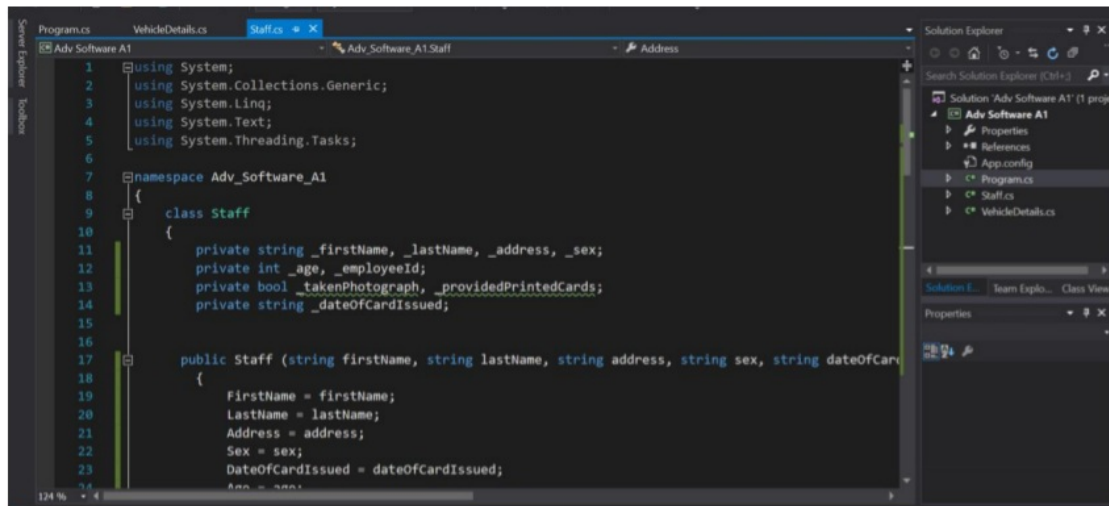
Object is and user defined and created by the keyword class, the nature of the object is defined by the class, the creation of new objects and managing them are done by the class. Group of similar objects are all contained in same class in the program (Rouse, 2005) (Anon., 2015).



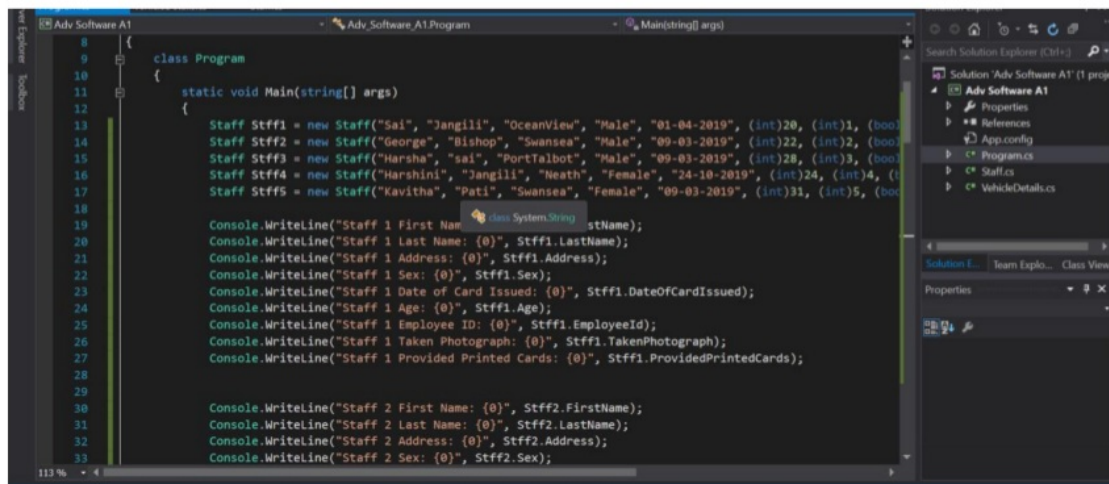
Difference between Class and an Object: -

Class	Object
It is prototype of an object that contains variables in a program.	Object is an occurrence of a class.
Class is a logical existence.	Object is physical existence.
No memory allocation is done during the runtime.	Required memory allocation is done at the time of declaration.
Class declaration is only once in a program	Object can be created as per the requirement.
Class can exist without any objects in it.	Objects can't exist without a class.
Class doesn't have any value.	Every object in the class has a value.
Class cannot be deployed	Object can be deployed.

(Anon., n.d.)



Here in the above image it shows the class in the program. The class is named as `staff`.



Here for the class `Staff` the objects are `Stff1`, `Stff2`, `Stff3`, `Stff4`, `Stff5`.

Data Encapsulation

The term data encapsulation is a method which is used to keep the details of the class hidden from the user. It restricts the direct access to few objects in the class.

components form user. It is also known as data hiding. An attribute which is not visible from an object will be bundled with the different types of methods and read and write access will be given to the attribute so that the specific information is hidden so that it can't be seen by the user. When the information is required by the user it can't get the access directly to get the data it will need to contact specific attribute to get the data stored in it. The importance of implementing data encapsulation occurs when the class implementation changes but the interface is same, in this case the user should use different methods to get the data from the attribute (craig, 1996) (JANSSEN, 2017) (Agarwal, n.d.).

Interfaces

An interface is like the class in a program but the difference between interface and class is interface clearly explains the behavior and the understanding between the user and an object. Interface is an agreement that a class should fulfill. Interface explains what a class must do and what not to do, it sends the instructions to the class in a program. Multiple inheritance can be occurred with interface but not with class. Interface explains the properties and procedures that should be available in class (Anon., 2017) (Anon., n.d.) (Microsoft, 2018).

Impact of interface in developing a code

Interface provides a true component-based programs, loose link which helps implementation getting easily separated from the interface. Using interface is easy to maintain and allows code to be reused and much more available because execution gets separated easily from the user interface. Using the interface in the code helps the code developers to describe the link between the different elements and interfaces clearly explains the responsibilities and the limitations between various elements (Cochran, 2019) (Berry, 2011)

References

Agarwal, H., n.d. *GeeksforGeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/encapsulation-in-c/>

[Accessed 16 FEB 2020].

Anon., 2015. *Microsoft*. [Online]

Available at: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/objects>

[Accessed 17 FEB 2020].

Anon., 2017. *Wiki Books*. [Online]

Available at: https://en.wikibooks.org/wiki/C_Sharp_Programming/Interfaces

[Accessed 16 FEB 2020].

Anon., n.d. *Java T point*. [Online]

Available at: <https://www.javatpoint.com/difference-between-object-and-class>

[Accessed 16 FEB 2020].

Anon., n.d. *TutorialsTeacher*. [Online]

Available at: <https://www.tutorialsteacher.com/csharp/csharp-interface>

[Accessed 16 FEB 2020].

Anon., n.d. *What is the Difference between a Classes and an Object?*. [Online]

Available at: <https://www.w3schools.in/java-questions-answers/difference-between-classes-objects/>

[Accessed 16 Feb 2020].

Berry, D., 2011. *redgate hub*. [Online]

Available at: <https://www.red-gate.com/simple-talk/dotnet/net-framework/designing-c-software-with-interfaces/>

[Accessed 16 FEB 2020].

Cochran, M., 2019. *C# Interface based development*. [Online]

Available at: <https://www.c-sharpcorner.com/article/C-Sharp-interface-based-development/>

[Accessed 16 FEB 2020].

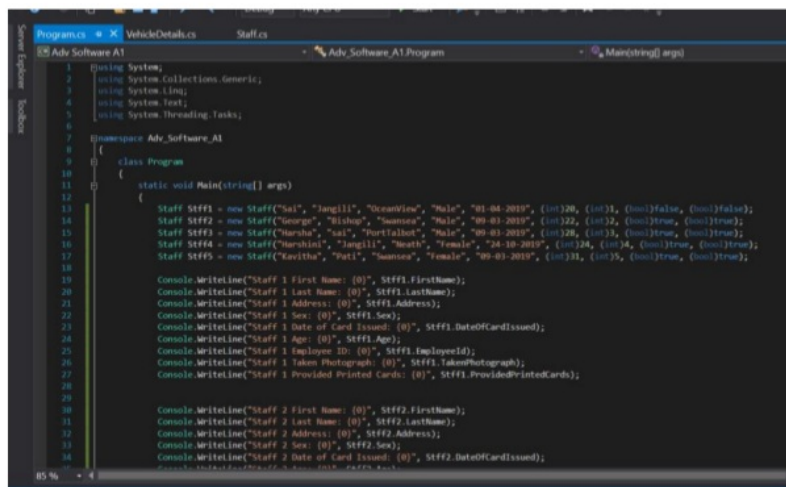
craig, D., 1996. *Data Encapsulation*. [Online]
Available at: <http://www.cs.mun.ca/~donald/bsc/node13.html>
[Accessed 17 FEB 2020].

JANSSEN, T., 2017. *Stackify*. [Online]
Available at: <https://stackify.com/oop-concept-for-beginners-what-is-encapsulation/>
[Accessed 16 FEB 2020].

Microsoft, 2018. *Microsoft*. [Online]
Available at: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/interfaces/>
[Accessed 16 FEB 2020].

Rouse, M., 2005. *TechTarget*. [Online]
Available at: <https://searchapparchitecture.techtarget.com/definition/object>
[Accessed 16 FEB 2020].

Appendices



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Adv_Software_A1
{
    class Program
    {
        static void Main(string[] args)
        {
            Staff Staff1 = new Staff("Jai", "Jangill", "Gowind", "Male", "01-04-2019", (int)120, (int)31, (bool>false, (bool>false);
            Staff Staff2 = new Staff("Suresh", "Bishup", "Suresh", "Male", "08-01-2019", (int)122, (int)32, (bool>true, (bool>true);
            Staff Staff3 = new Staff("Suresh", "Jai", "Porttallo", "Male", "09-01-2019", (int)124, (int)33, (bool>true, (bool>true);
            Staff Staff4 = new Staff("Suresh", "Jangill", "Suresh", "Female", "24-10-2019", (int)124, (int)34, (bool>true, (bool>true);
            Staff Staff5 = new Staff("Kavitha", "Patil", "Suresh", "Female", "09-01-2019", (int)131, (int)35, (bool>true, (bool>true);

            Console.WriteLine("Staff 1 First Name: {0}", Staff1.FirstName);
            Console.WriteLine("Staff 1 Last Name: {0}", Staff1.LastName);
            Console.WriteLine("Staff 1 Address: {0}", Staff1.Address);
            Console.WriteLine("Staff 1 Sex: {0}", Staff1.Sex);
            Console.WriteLine("Staff 1 Date of Card Issued: {0}", Staff1.DateOfCardIssued);
            Console.WriteLine("Staff 1 Age: {0}", Staff1.Age);
            Console.WriteLine("Staff 1 Employee ID: {0}", Staff1.EmployeeID);
            Console.WriteLine("Staff 1 Taken Photograph: {0}", Staff1.TakenPhotograph);
            Console.WriteLine("Staff 1 Provided Printed Cards: {0}", Staff1.ProvidedPrintedCards);

            Console.WriteLine("Staff 2 First Name: {0}", Staff2.FirstName);
            Console.WriteLine("Staff 2 Last Name: {0}", Staff2.LastName);
            Console.WriteLine("Staff 2 Address: {0}", Staff2.Address);
            Console.WriteLine("Staff 2 Sex: {0}", Staff2.Sex);
            Console.WriteLine("Staff 2 Date of Card Issued: {0}", Staff2.DateOfCardIssued);
```

```

Program.cs VehicleDetails.cs Staff.cs
Adv Software A1
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Adv_Software_A1
{
    class Staff
    {
        private string _firstName, _lastName, _address, _sex;
        private int _age, _employeeId;
        private bool _takenPhotograph, _providedPrintedCards;
        private string _dateOfCardIssued;

        public Staff(string firstName, string lastName, string address, string sex, string dateOfCardIssued, int age, int employeeId, bool takenPhotograph, bool providedPrintedCards)
        {
            _firstName = firstName;
            _lastName = lastName;
            _address = address;
            _sex = sex;
            _dateOfCardIssued = dateOfCardIssued;
            _age = age;
            _employeeId = employeeId;
            _takenPhotograph = takenPhotograph;
            _providedPrintedCards = providedPrintedCards;
        }

        public string FirstName
        {
            get { return _firstName; }
            set { _firstName = value; }
        }

        public string LastName
        {
            get { return _lastName; }
            set { _lastName = value; }
        }

        public string Address
        {
            get { return _address; }
            set { _address = value; }
        }

        public string Sex
        {
            get { return _sex; }
            set { _sex = value; }
        }

        public string DateOfCardIssued
        {
            get { return _dateOfCardIssued; }
            set { _dateOfCardIssued = value; }
        }

        public int Age
        {
            get { return _age; }
            set { _age = value; }
        }

        public int EmployeeId
        {
            get { return _employeeId; }
            set { _employeeId = value; }
        }

        public bool TakenPhotograph
        {
            get { return _takenPhotograph; }
            set { }
        }

        public bool ProvidedPrintedCards
        {
            get { return _providedPrintedCards; }
            set { }
        }
    }
}

```

```

Program.cs VehicleDetails.cs Staff.cs
Adv Software A1
public string FirstName
{
    get { return _firstName; }
    set { _firstName = value; }
}

public string LastName
{
    get { return _lastName; }
    set { _lastName = value; }
}

public string Address
{
    get { return _address; }
    set { _address = value; }
}

public string Sex
{
    get { return _sex; }
    set { _sex = value; }
}

public string DateOfCardIssued
{
    get { return _dateOfCardIssued; }
    set { _dateOfCardIssued = value; }
}

public int Age
{
    get { return _age; }
    set { _age = value; }
}

public int EmployeeId
{
    get { return _employeeId; }
    set { _employeeId = value; }
}

public bool TakenPhotograph
{
    get { return _takenPhotograph; }
    set { }
}

public bool ProvidedPrintedCards
{
    get { return _providedPrintedCards; }
    set { }
}

```

```

Program.cs VehicleDetails.cs Staff.cs
Adv Software A1
Staff staff1 = new Staff("George", "Bishop", "Sussex", "Male", "01-01-2000", (int)12, (int)12, (bool)true, (bool)true);
Staff staff2 = new Staff("Nurula", "Ali", "Portlinton", "Male", "01-01-2000", (int)18, (int)13, (bool)true, (bool)true);
Staff staff3 = new Staff("Nurulana", "Bangli", "Nuch", "Female", "26-10-2000", (int)14, (int)14, (bool)true, (bool)true);
Staff staff4 = new Staff("Kavitha", "Paul", "Sussex", "Female", "01-01-2000", (int)18, (int)14, (bool)true, (bool)true);

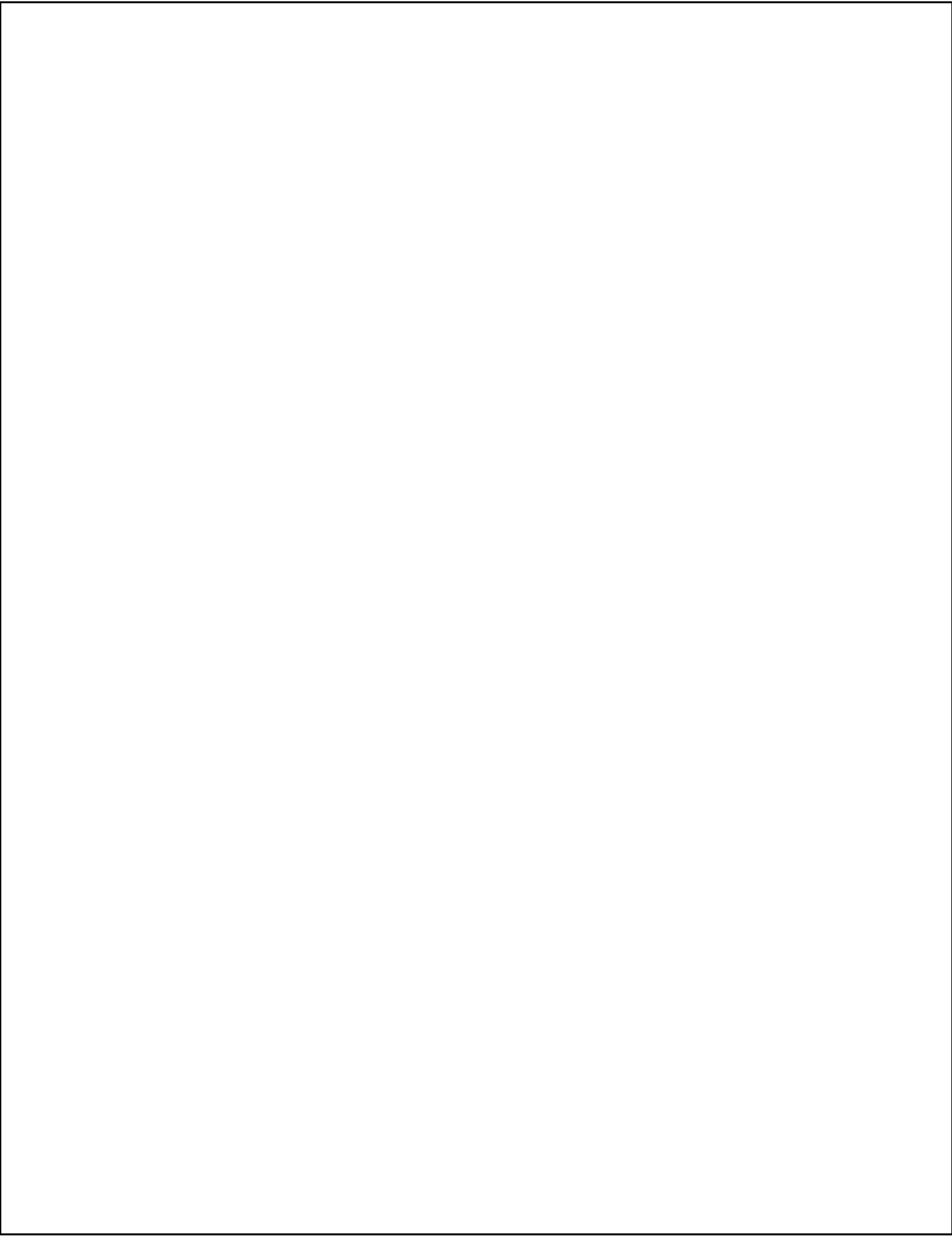
Console.WriteLine("Staff 1 First Name: {0}", staff1.FirstName);
Console.WriteLine("Staff 1 Last Name: {0}", staff1.LastName);
Console.WriteLine("Staff 1 Address: {0}", staff1.Address);
Console.WriteLine("Staff 1 Sex: {0}", staff1.Sex);
Console.WriteLine("Staff 1 Date of Card Issued: {0}", staff1.DateOfCardIssued);
Console.WriteLine("Staff 1 Age: {0}", staff1.Age);
Console.WriteLine("Staff 1 Employee ID: {0}", staff1.EmployeeId);
Console.WriteLine("Staff 1 Taken Photograph: {0}", staff1.TakenPhotograph);
Console.WriteLine("Staff 1 Provided Printed Cards: {0}", staff1.ProvidedPrintedCards);

Console.WriteLine("Staff 2 First Name: {0}", staff2.FirstName);
Console.WriteLine("Staff 2 Last Name: {0}", staff2.LastName);
Console.WriteLine("Staff 2 Address: {0}", staff2.Address);
Console.WriteLine("Staff 2 Sex: {0}", staff2.Sex);
Console.WriteLine("Staff 2 Date of Card Issued: {0}", staff2.DateOfCardIssued);
Console.WriteLine("Staff 2 Age: {0}", staff2.Age);
Console.WriteLine("Staff 2 Employee ID: {0}", staff2.EmployeeId);
Console.WriteLine("Staff 2 Taken Photograph: {0}", staff2.TakenPhotograph);
Console.WriteLine("Staff 2 Provided Printed Cards: {0}", staff2.ProvidedPrintedCards);

Console.WriteLine("Staff 3 First Name: {0}", staff3.FirstName);
Console.WriteLine("Staff 3 Last Name: {0}", staff3.LastName);
Console.WriteLine("Staff 3 Address: {0}", staff3.Address);
Console.WriteLine("Staff 3 Sex: {0}", staff3.Sex);
Console.WriteLine("Staff 3 Date of Card Issued: {0}", staff3.DateOfCardIssued);
Console.WriteLine("Staff 3 Age: {0}", staff3.Age);
Console.WriteLine("Staff 3 Employee ID: {0}", staff3.EmployeeId);
Console.WriteLine("Staff 3 Taken Photograph: {0}", staff3.TakenPhotograph);
Console.WriteLine("Staff 3 Provided Printed Cards: {0}", staff3.ProvidedPrintedCards);

Console.WriteLine("Staff 4 First Name: {0}", staff4.FirstName);
Console.WriteLine("Staff 4 Last Name: {0}", staff4.LastName);
Console.WriteLine("Staff 4 Address: {0}", staff4.Address);
Console.WriteLine("Staff 4 Sex: {0}", staff4.Sex);

```



FINAL GRADE

31

/100

GENERAL COMMENTS

InstructorPart 1 :

Class (10): 7

Encapsulation (10): 8

Interface (10): 8

Part 2:

Class (10): 7

Logic (10): 1

Polymorphism (30): 0

Part 3:

Written (10):

Code (10):

Comments

=====

Some of the discussion of the concepts is a bit debatable, although the key ideas are presented.

Part 2:

Code is very basic, just a couple of classes with constructors and properties. No inheritance, polymorphism or the logic required,

PAGE 1

PAGE 2

PAGE 3

**Comment 1**

Must have a valid reference



Comment 2

"done by" implies something that is not true



Comment 3

How is this relevant to the question?

PAGE 4



Comment 4

Poor wording...."attributes"

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9
