Lab 04

Object-Oriented Programming

- Write a class “Student” which stores student info such as: name, email, phone, major and university
- Define multiple constructors that accept different number/types of parameters
- Write a method that counts the number of instances created from this class.
- Write methods to get/set different pieces of info for students
- Write a method to print all student info
- Test your class and methods

Student.cs

```
   class Student
    {

        string name, email, university, major;
        long phone;

        //a static field that is used to count the number of instances
created from this class
        static int count = 0;

        /// <summary>
        /// Student class constructor, initializes all five data fields
using acquired parameters
        /// </summary>
        /// <param name="n">Student's name</param>
        /// <param name="e">Student's email</param>
        /// <param name="u">Student's university</param>
        /// <param name="m">Student's major</param>
        /// <param name="p">Studnent's phone number</param>
        public Student(string n, string e, string u, string m, long p)
        {
            this.name = n;
            this.email = e;
            this.university = u;
            this.major = m;
            this.phone = p;
            count++;
        }

        /// <summary>
        /// Student class constructor, intializes student name to the
received string value and sets all other fields to pre-defined default
values
        /// </summary>
        /// <param name="n">Student's name</param>
        public Student(string n)
        {
            this.name = n;
            this.email = "No email assigned yet";
            this.university = "No university assigned yet";
            this.major = "No major assigned yet";
            this.phone = 0;
            count++;
        }
```

```
        /// <summary>
        /// A method to get student name
        /// </summary>
        /// <returns>Returns student name (string)</returns>
        public string GetName()
        {
            return this.name;
        }

        /// <summary>
        /// A method to get student's email
        /// </summary>
        /// <returns>Returns student email (string)</returns>
        public string GetEmail()
        {
            return this.email;
        }

        /// <summary>
        /// A method to get student's university
        /// </summary>
        /// <returns>Returns studnet's university (string)</returns>
        public string GetUniversity()
        {
            return this.university;
        }

        /// <summary>
        /// A method to get student's major
        /// </summary>
        /// <returns>Returns student's major (string)</returns>
        public string GetMajor()
        {
            return this.major;

        }

        /// <summary>
        /// A method to get student's phone number
        /// </summary>
        /// <returns>Returns student's phone number (long)</returns>
        public long GetPhone()
        {
            return this.phone;
        }
```

```
        /// <summary>
        /// A method that changes student's name to a new value
        /// </summary>
        /// <param name="newName"><c>string</c> The new value for
student's name</param>
        public void SetName(string newName)
        {
            this.name = newName;
        }

        /// <summary>
        /// A method that changes student's email to a new value
        /// </summary>
        /// <param name="newEmail"><c>string</c>The new value for
student's email</param>
        public void SetEmail(string newEmail)
        {
            this.email = newEmail;
        }

        /// <summary>
        /// A method that changes student's major to a new value
        /// </summary>
        /// <param name="newMajor"><c>string</c> The new value for
student's major</param>
        public void SetMajor(string newMajor)
        {
            this.major = newMajor;
        }

        /// <summary>
        /// A method that changes student's university to a new value
        /// </summary>
        /// <param name="newUni"><c>string</c> The new value for
student's university</param>
        public void SetUniversity(string newUni)
        {
            this.university = newUni;
        }


        /// <summary>
        /// A method that changes student's phone to a new value
        /// </summary>
        /// <param name="newPhone"><c>long</c> The new value for
student's phone</param>
        public void SetPhone(long newPhone)
        {
            this.phone = newPhone;
        }
```

```
        /// <summary>
        /// A static method counts the number of instances created from
the <c>Student</c> class
        /// </summary>
        /// <returns>Returns an <c>int</c> representing the number of
instances created from the <c>Student</c> class</returns>
        public static int StudentsCount()
        {
            return count;
        }

        /// <summary>
        /// A method that prints all <c>Student</c>'s data fields
values
        /// </summary>
        public void PrintInfo()
        {
            Console.WriteLine("Name: {0}\nEmail: {1}\nMajor:
{2}\nUniversity: {3}\nPhone: {4}", this.name, this.email, this.major,
this.university, this.phone);
        }
    }
```

Tester Class (Program.cs)

```
   class Program
    {
        static void Main(string[] args)
        {
            //find the number of created student objects, should be
zero now
            Console.WriteLine("Number of students created: {0}",
Student.StudentsCount());

            //create a new student s1 , providing all five parameters
for the first overloaded constructor
            Student s1 = new Student("Ahmed", "ahmed@mans.edu.eg",
"Mansoura", "Computer Engineering", 010000000);

            //print all s1 info
            Console.WriteLine("s1 Info:");
            s1.PrintInfo();

            //change s1 email
            s1.SetEmail("ahmed.ali@gmail.com");

            //now, print s1 email to make sure it has been successfully
changed
            Console.WriteLine("s1 email now is: {0}", s1.GetEmail());

            //find the number of objects created, should be one now
            Console.WriteLine("Number of students created: {0}",
Student.StudentsCount());

            //create a new student s2 using the second version of the
overloaded constructor which takes only a string representing the name
            Student s2 = new Student("Mahmoud");

            //print s2 info
            Console.WriteLine("s2 Info:");
            s2.PrintInfo();

            //change s2 field values (other than name) to override
default values given by the constructor
            s2.SetEmail("mahmoud@yahoo.com");
            s2.SetMajor("Civil Engineering");
            s2.SetPhone(0120000000);
            s2.SetUniversity("Cairo");

            //print s2 info again after changing its fields values
            Console.WriteLine("s2 Info:");
            s2.PrintInfo();
```

```
            //finally, find the number of created objects from Student
class
            Console.WriteLine("Number of students created: {0}",
Student.StudentsCount());

        }
    }
```