Object Orientated Programming

Lab Manual 11

Instructor:

Mr. Samyan Qayyum Wahla

Learning Objectives:

- Student should be able to understand and define functions
- Learn to write well documented and formatted code.

CLO:

- CLO1

Registration Number:

_____

Name:

_____

Guidelines/Instructions:

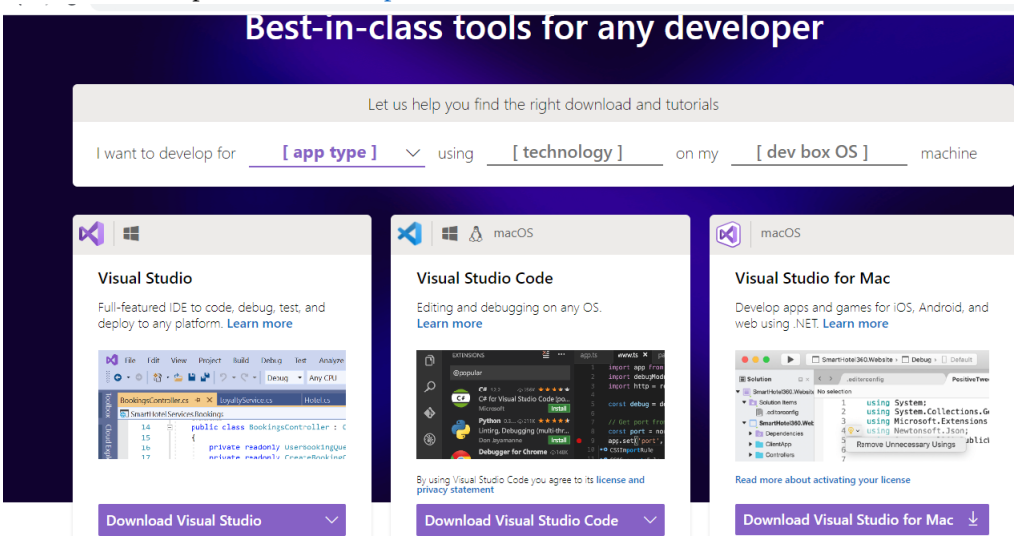
- Use of visual studio is must in this lab.
- Write well commented code.
- Name of variables should be meaningful.
- Code should be well formatted.
- Create meaningful variable names. Add comments for readability. Indent each line of your code.
- Plagiarism/Cheating is highly discouraged by penalizing to both who tried and one who shared his/her code.

Today's Task

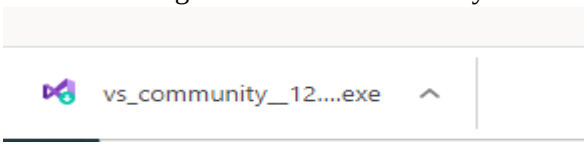
- Download visual studio
- Install .Net Desktop Development Framework
- Write Hello World Program in C#
- Java to C#

Installation Guide

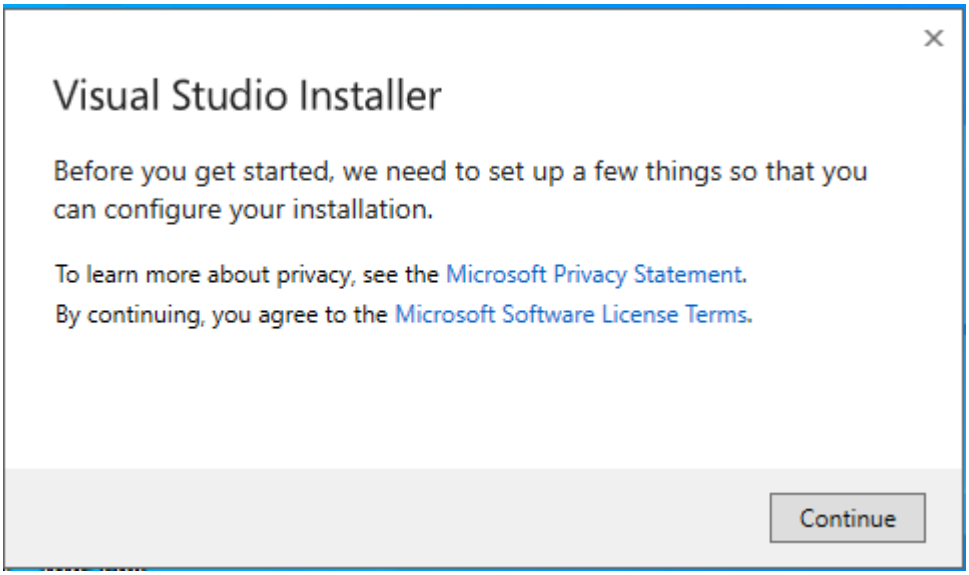
1. Download visual studio from provided url: https://visualstudio.microsoft.com/



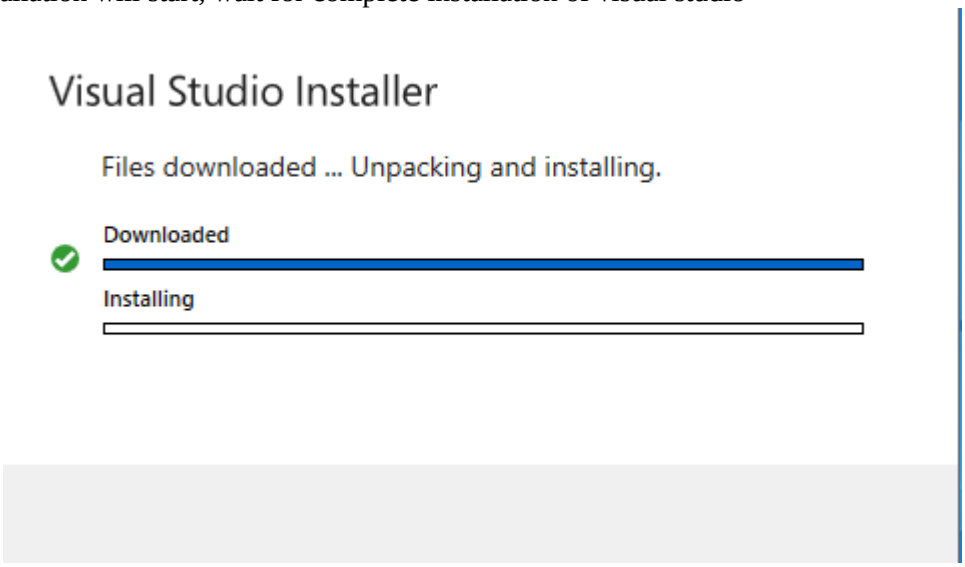
2. Click on Download Visual Studio in left most grid and select community 2019 setup for installation



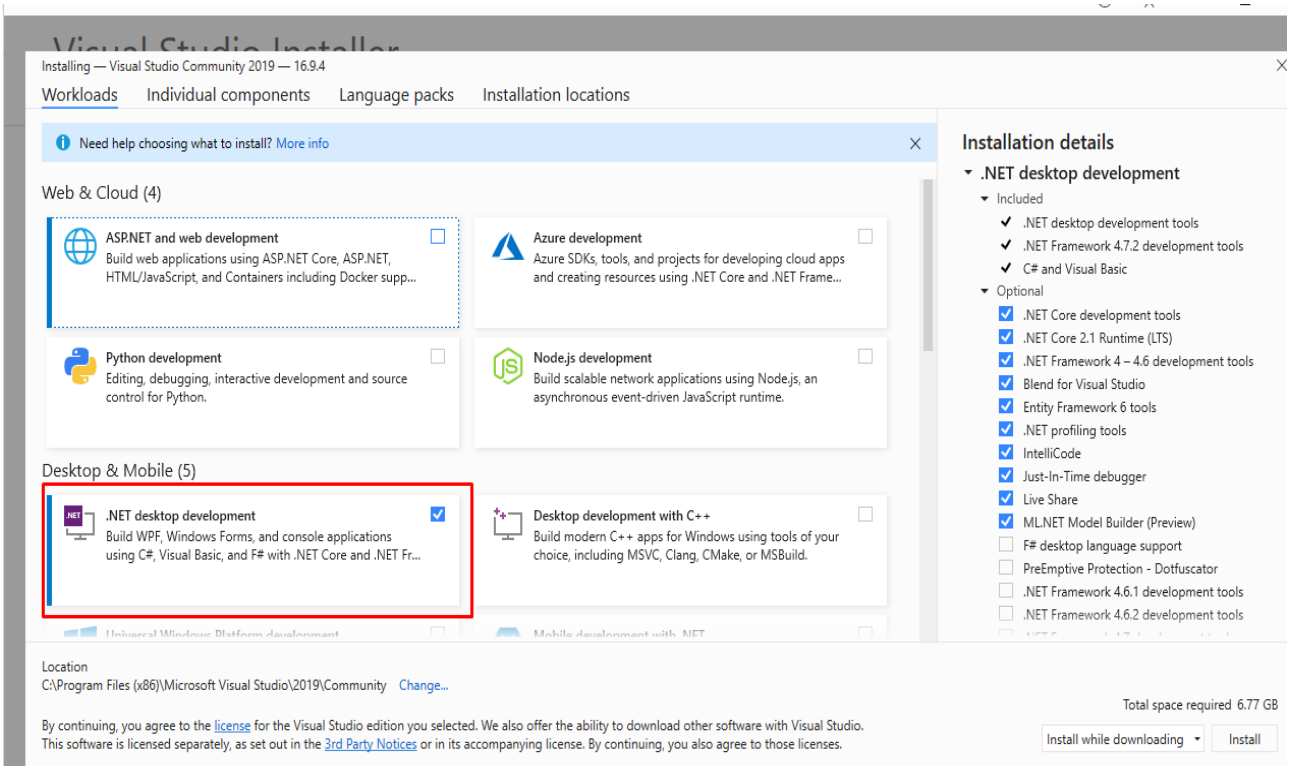
3. Run the downloaded executable file , allow it to run as administrator, an installer window will open to continue the installation

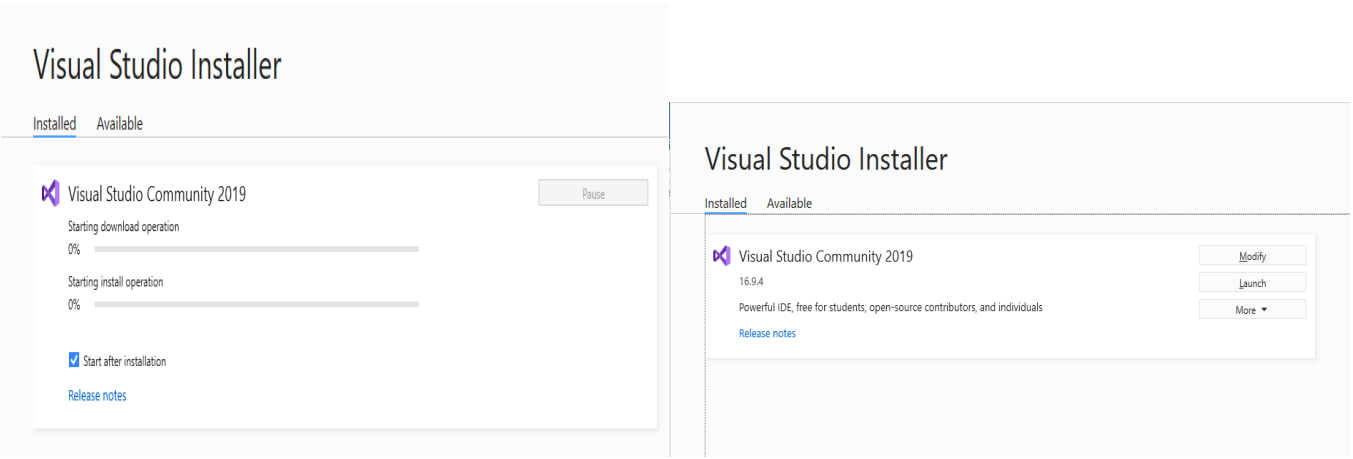


4. After that the installation will start, wait for complete installation of visual studio

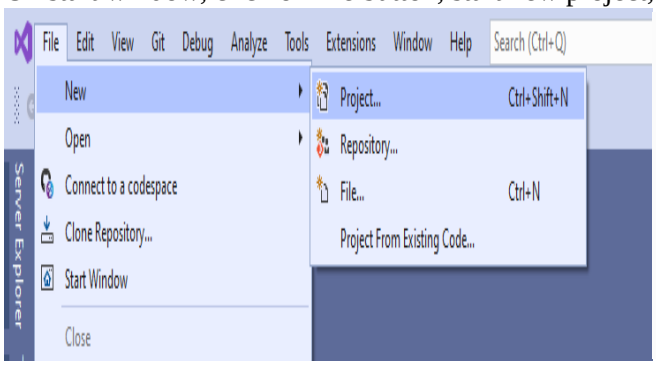


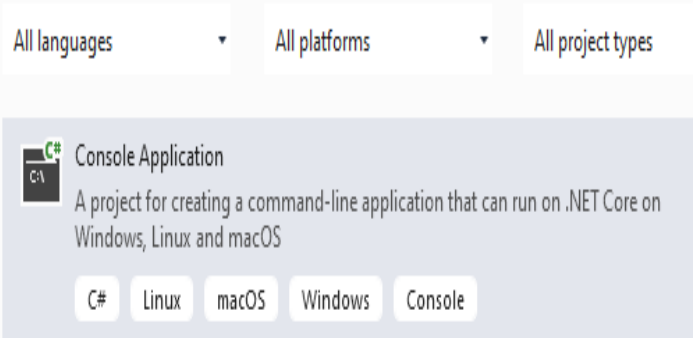
5. On complete installation, workload window will open. Select the ".Net desktop Development" package and click on install button in button right corner to proceed. Wait for complete installation and click on Launch button to start.



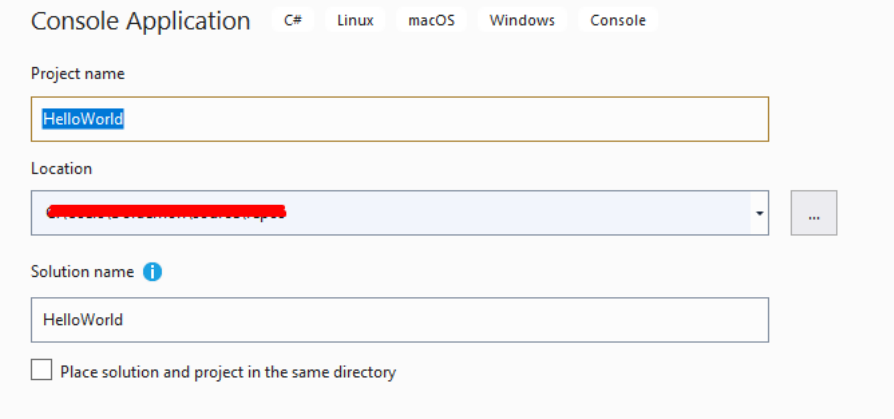


6. On start window, click on file button, start new project, and select the Console Application.

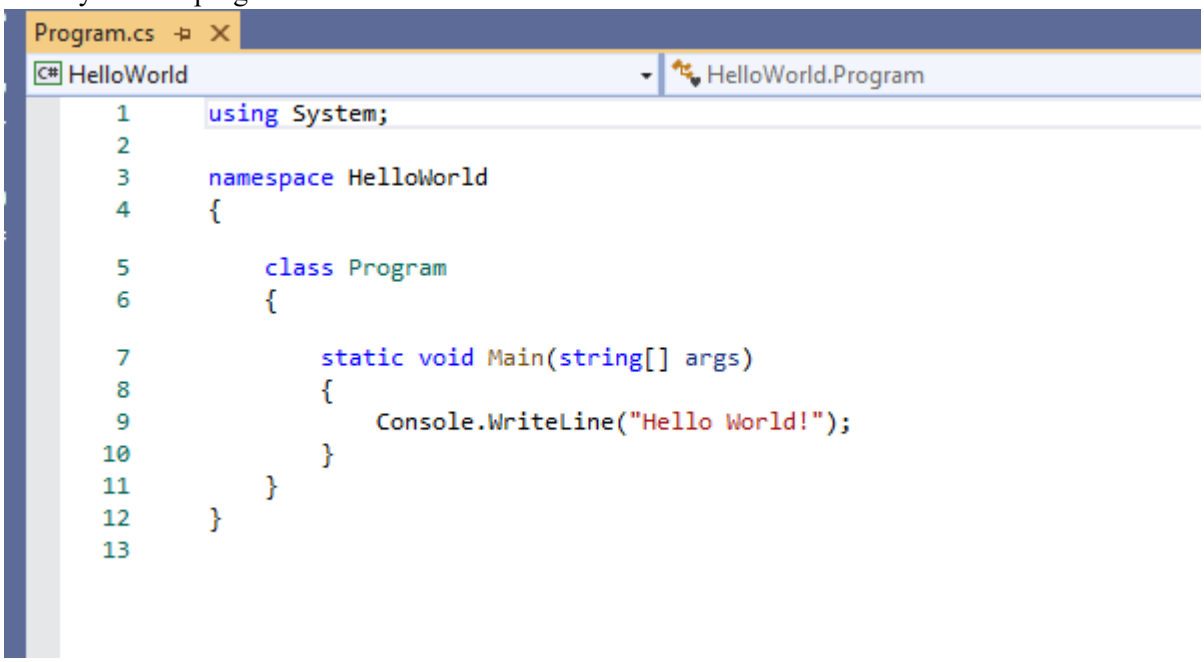




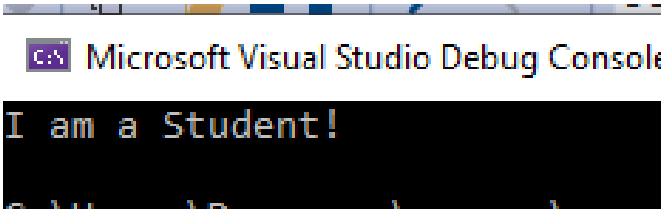
7. Write the file name



8. Write your first program on visual studio IDE and click on run button on "Run" button.



The output of above program will be



Examples:

Example 1: Write a Program to display integer value

C#	Java Code

using System;
namespace displayNumber
{
class Program
{
static void Main(string[] args)
{
Console.WriteLine("Welcome
To Program! \n");
int num = 70;
	public class FirstProgram {
	
	public static void main(String [] args)
	//main method for program execution
	{
	int num = 70;
	System.out.println("Welcome to Object Orientated
	Programming");

```
            Console.WriteLine("Number is: "
            + num);
            Console.ReadLine();
        }
    }
}
```

```
    System.out.println("Number Entered is: " + num);
  }
}
```

Example 2: Input for Integer

Write a program that ask user to enter age and display it

C#

```
using System;
namespace inputAge
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Welcome to Input Program!");
            Console.WriteLine("Enter Your Age");
            int age;
            age = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Your Age is: " + age);
        }
    }
}
```

Example 3: Input for Double

Write a program to calculate area of triangle

C#

```
using System;
namespace triangle
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Welcome to Input Program!");

            double height, width, area;
            Console.WriteLine("Enter Height");
            height = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine("Enter Width");
            width = Convert.ToDouble(Console.ReadLine());
            area = height * width;
            Console.WriteLine("Area is: " + area);
        }
    }
}
```

Example 4: Input for String
Write a program that display string input to user.

C#

```
using System;
namespace displayString
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Welcome to Input Program!");
            String name;
            Console.WriteLine("Enter Name");
            name = Console.ReadLine();
            Console.WriteLine("Name is: " + name);
        }
    }
}
```

Example 5: Conditional Statements
Write a program that display a number if value entered by user is positive integer

C#

```
using System;
namespace displayInt
{
    class Program
    {
        static void Main(string[] args)
        {
            int number;
            Console.WriteLine("Enter Name");
            number = Convert.ToInt32(Console.ReadLine());
            if (number > 0)
            {
                Console.WriteLine("Number is: " + number);
            }
            else
            {
                Console.WriteLine("Enter Positive Integer");
            }
        }
    }
}
```

Example 6: Loops
Write a program that print all even positive integers till n (input entered by user) using loop.

C#

```
using System;

namespace printEven
{
    class Program
    {
```

```
        static void Main(string[] args)
        {
            int number;
            Console.WriteLine("Enter Name");
            number = Convert.ToInt32(Console.ReadLine());
           for (int i =1;  i< number; i++)
            {
                if (i%2 == 0)
                {
                    Console.WriteLine(i + " ");
                }
            }
        }
    }
}
```

Now Explore C# for object orientated programming on your own

Task

Write a C# program in which you are required to define a class named **Student**. The class must include the following data members.

- **studentName** //data type should be String (name should be alphabetic, special characters and numbers are not allowed)
- **registrationNumber** //data type should be String (Format should be like this: 2020-CS-888, any other format should be handled in setter function)
- **dateOfBirth** //data type should be Date (Date of birth should be less than 1st January 2005 and greater than 31st December 1990, add conditions for invalid data)
- **cgpa** // data type should be double (should be between 0 and 4 inclusive)
- **cnic** //data type should be String (only numbers are allowed with valid CNIC)
- **numberOfInstances** //This attribute should track the number of objects of the class Student. (Make it static. Don't create setter for this attribute)

1. Define getter setter for each data member
2. Apart from getter, setter and constructor, define the following function

   i. **String getAge()** – it should calculate age of student upto current date format given in output section.
   ii. **String getStatus()** – function should tell the status of student in the following cases
      a. IF CGPA less than 2.0 - Status is Suspended
      b. IF CGPA between 2.0 and 2.5 - Status is Below Average
      c. IF CGPA between 2.5 and 3.3 - Status is Average
      d. IF CGPA between 3.3 and 3.5 - Status is Below Good
      e. IF CGPA greater than 3.5 – Status is Excellent
   iii. **int numberOfWordsInName()** – it should return number of words in students name
   iv. **String getGender()** – if last digit of CNIC is odd then it should return MALE otherwise FEMALE
   v. **String toString() –** Display the output in specific format given in output
   vi. **void input() –** Take the input from the user in this method. You can read input from user using InputDialog or Scanner.

   Declare two objects of Student type in **Driver class**.

Output

Your output should be like the following with same formatting
Name: Sarfraz Ahmed (Contain 2 words)
Registration Number: 2018-CS-999
CGPA: 2.9 Average
Date of Birth: December 27, 1999 (Age is 21 years 8 months and 12 days)
CNIC: 3567912356781
Gender: Male

Name: Hafiz Muhammad Aslam (Contain 3 words)
Registration Number: 2017-CS-999
CGPA: 3.4 Good
Date of Birth: November 27, 1999 (Age is 21 years 9 months and 12 days)
CNIC: 3567912356785
Gender: Male

Student class has the 2 objects

What to submit

You are simply required to submit a source file (**Student.cs** and **Driver.cs** (in which main function is defined)) that includes the implementation of the above mentioned program. No extra file should be submitted.