4.3 Inheritance

This section will guide you to:

* Create a Windows Console project in Visual Studio to demonstrate the use of Inheritance in Classes
* Create classes for CClass and Teacher
* Create inherited classes for SubjectTeacher and ClassTeacher

**Development Environment**

* Visual Studio 2019 Community Version
* Windows 10

This guide has nine subsections, namely:

* + 1. Creating a Windows Console project in Visual Studio to demonstrate the use of inheritance in classes
    2. Creating a CClass class to store data about teachers in a class
    3. Creating a Teacher class to store teachers data
    4. Creating a ClassTeacher class to store data about class teachers
    5. Creating a SubjectTeacher class to store data about subject teachers
    6. Adding a method, runApp(), in Program class to create objects and populate them
    7. Building the project
    8. Publishing and running the project
    9. Pushing the code to your GitHub repositories

**Step 4.3.1:** Creating a Windows Console project in Visual Studio to demonstrate the use of inheritance in classes

* Open Visual Studio.
* From the top menu, select **File->New->Project.**
* In **Create A New Project** screen, select **Console app (.NET Core)** from the list of available project types and click on **Next.**
* Enter **Project Name** as **Phase1Section5.8** and click on **Create.**
* This will create the files for a Windows Console project.

**Step 4.3.2:** Creating a CClass class to store data about teachers in a class

* In the **Solution Explorer** window, right-click **Phase1Section5.8** and choose **Add->Class.**
* Enter Class name as **CClass.cs** and click **Add.**
* Enter the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Text;

**using** System.Collections;

**namespace** Phase1Section5.\_8

{

**public** **class** CClass

{

**private** **string** name;

**private** ClassTeacher whichTeacher;

**private** List<SubjectTeacher> subjectTeachers;

**public** **string** Name

{

**get** { **return** name; }

**set** { name = value; }

}

**public** ClassTeacher WhichTeacher

{

**get** { **return** whichTeacher; }

**set** { whichTeacher = value; }

}

**public** List<SubjectTeacher> SubjectTeachers

{

**get** { **return** subjectTeachers; }

**set** { subjectTeachers = value; }

}

}

}

**Step 4.3.3:** Creating a Teacher class to store data about teachers

* In the **Solution Explorer** window, right-click **Phase1Section5.8** and choose **Add->Class.**
* Enter Class name as **Teacher.cs** and click **Add.**
* Enter the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Text;

**namespace** Phase1Section5.\_8

{

**public** **class** Teacher

{

**private** **string** name;

**private** **string** address;

**private** **string** contactAddress;

**private** DateTime dateOfJoining;

**public** **string** Name

{

**get** { **return** name; }

**set** { name = value; }

}

**public** **string** ContactAddress

{

**get** { **return** address; }

**set** { address = value; }

}

**public** DateTime DateOfJoining

{

**get** { **return** dateOfJoining; }

**set** { dateOfJoining = value; }

}

}

}

**Step 4.3.4:** Creating a ClassTeacher class to store data about class teachers

* In the **Solution Explorer** window, right-click **Phase1Section5.8** and choose **Add->Class.**
* Enter Class name as **ClassTeacher.cs** and click **Add.**
* Enter the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Text;

**namespace** Phase1Section5.\_8

{

**public** **class** ClassTeacher: Teacher

{

**private** CClass whichClass;

**public** CClass WhichClass

{

**get** { **return** whichClass; }

**set** { whichClass = value; }

}

}

}

**Step 4.3.5:** Creating a SubjectTeacher class to store data about subject teachers

* In the **Solution Explorer** window, right-click **Phase1Section5.8** and choose **Add->Class.**
* Enter Class name as **SubjectTeacher.cs** and click **Add.**
* Enter the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Text;

**namespace** Phase1Section5.\_8

{

**public** **class** SubjectTeacher: Teacher

{

**private** Subject whichSubject;

**public** Subject WhichSubject

{

**get** { **return** whichSubject; }

**set** { whichSubject = value; }

}

}

}

**Step 4.3.6:** Adding a method, runApp(), in Program class to create objects and populate them

* Select **Program.cs** as the current Code tab.
* Enter the following code:

**using** System;

**namespace** Phase1Section5.\_8

{

**class** Program

{

**static** **void** Main(**string**[] args)

{

runApp();

}

**public** **static** **void** runApp()

{

Subject algebra = **new** Subject();

algebra.Name = "Algebra";

Subject physics = **new** Subject();

physics.Name = "Physics";

SubjectTeacher algebraTeacher = **new** SubjectTeacher();

algebraTeacher.Name = "Mr.Algebra";

algebraTeacher.ContactAddress = "Some Address";

algebraTeacher.DateOfJoining = Convert.ToDateTime("2009-10-20 00:00:00");

algebraTeacher.WhichSubject = algebra;

SubjectTeacher physicsTeacher = **new** SubjectTeacher();

physicsTeacher.Name = "Mr.Physics";

physicsTeacher.ContactAddress = "Some Address";

physicsTeacher.DateOfJoining = Convert.ToDateTime("2009-10-20 00:00:00");

physicsTeacher.WhichSubject = physics;

CClass class8 = **new** CClass();

class8.Name = "8";

ClassTeacher teacherOf8 = **new** ClassTeacher();

teacherOf8.Name = "Mrs.Class 8 Teachers";

teacherOf8.ContactAddress = "Some Address";

teacherOf8.DateOfJoining = Convert.ToDateTime("2009-10-20 00:00:00");

teacherOf8.WhichClass = class8;

class8.WhichTeacher = teacherOf8;

}

}

}

**Step 4.3.7:** Building the project

* From the top menu, choose **Build->Build Solution.**
* If any compile errors are shown, fix them as required.

**Step 4.3.8:** Publishing and running the project

* From the top menu, select **Debug->Start Without Debugging.**
* This will execute the program in a console window. Since there is no output, nothing will be displayed.

**Step 4.3.9:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit -m “Changes have been committed.”

Push the files to the folder you created initially using the following command:

git push -u origin master