4.4 Polymorphism 

This section will guide you to:

* Create a Windows Console project in Visual Studio to demonstrate the use of polymorphism in Classes
* Create class for basic grade card of a student
* Create inherited classes for grade cards of primary, middle and high school students using polymorphism

**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 polymorphism in classes
    2. Creating a BasicGradeCard class to act as a template for grade cards
    3. Creating an ElementarySchoolGradeCard class derived from BasicGradeCard for primary school students
    4. Creating a MiddleSchoolGradeCard class derived from BasicGradeCard for middle school students
    5. Creating a HighSchoolGradeCard class derived from BasicGradeCard for highschool students
    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 4.4.1:** Creating a Windows Console project in Visual Studio to demonstrate the use of classes

* Open Visual Studio.
* From the top menu, select **File->New->Project.**
* In the **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 Phase1Section5.10 and click on **Create.**
* This will create the files for a Windows Console project.

**Step 4.4.2:** Creating a BasicGradeCard class to act as a template for grade cards.

* In the **Solution Explorer** window, right click **Phase1Section5.10** and choose **Add->Class.**
* Enter class name as BasicGradeCard.cs and click **Add.**
* Enter the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Text;

**namespace** Phase1Section5.\_10

{

**public** **class** BasicGradeCard

{

**protected** **int** totalMarks;

**public** BasicGradeCard()

{

totalMarks = 0;

}

**public** **virtual** **int** getTotalMarks()

{

**return** totalMarks;

}

}

}

**Step 4.4.3:** Creating an ElementarySchooGradeCard class derived from BasicGradeCard for primary school students

* In the **Solution Explorer** window, right click **Phase1Section5.10** and choose **Add->Class.**
* Enter Class name as ElementarySchoolGradeCard.cs and click **Add.**
* Enter the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Text;

**namespace** Phase1Section5.\_10

{

**public** **class** ElementarySchoolGradeCard: BasicGradeCard

{

**private** **int** maths = 0;

**private** **int** english = 0;

**private** **int** secondLanguage = 0;

**private** **int** socialScience = 0;

**public** **int** Maths {

**get** { **return** maths; }

**set** { maths = value; }

}

**public** **int** English

{

**get** { **return** english; }

**set** { english = value; }

}

**public** **int** SecondLanguage

{

**get** { **return** secondLanguage; }

**set** { secondLanguage = value; }

}

**public** **int** SocialScience

{

**get** { **return** socialScience; }

**set** { socialScience = value; }

}

**public** **virtual** **int** getTotalMarks()

{

**this**.totalMarks = maths + english + socialScience + secondLanguage;

**return** totalMarks;

}

}

}

**Step 4.4.4:** Creating a MiddleSchoolGradeCard class derived from BasicGradeCard for middle school students

* In the **Solution Explorer** window, right click **Phase1Section5.10** and choose **Add->Class.**
* Enter Class name as MiddleSchoolGradeCard.cs and click **Add.**
* Enter the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Text;

**namespace** Phase1Section5.\_10

{

**public** **class** MiddleSchoolGradeCard : BasicGradeCard

{

**private** **int** maths = 0;

**private** **int** english = 0;

**private** **int** secondLanguage = 0;

**private** **int** geography = 0;

**private** **int** history = 0;

**public** **int** Maths

{

**get** { **return** maths; }

**set** { maths = value; }

}

**public** **int** English

{

**get** { **return** english; }

**set** { english = value; }

}

**public** **int** SecondLanguage

{

**get** { **return** secondLanguage; }

**set** { secondLanguage = value; }

}

**public** **int** Geography

{

**get** { **return** geography; }

**set** { geography = value; }

}

**public** **int** History

{

**get** { **return** history; }

**set** { history = value; }

}

**public** **virtual** **int** getTotalMarks()

{

**this**.totalMarks = maths + english + geography + history;

**return** totalMarks;

}

}

}

**Step 4.4.5:** Creating a HighSchoolGradeCard class derived from BasicGradeCard for highschool students

* In the **Solution Explorer** window, right click **Phase1Section5.10** and choose **Add->Class.**
* Enter Class name as HighSchoolGradeCard.cs and click **Add**.
* Enter the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Text;

**namespace** Phase1Section5.\_10

{

**public** **class** HighSchoolGradeCard : BasicGradeCard

{

**private** **int** maths = 0;

**private** **int** english = 0;

**private** **int** secondLanguage = 0;

**private** **int** geography = 0;

**private** **int** history = 0;

**private** **int** physics = 0;

**private** **int** chemistry = 0;

**private** **int** biology = 0;

**public** **int** Maths

{

**get** { **return** maths; }

**set** { maths = value; }

}

**public** **int** English

{

**get** { **return** english; }

**set** { english = value; }

}

**public** **int** SecondLanguage

{

**get** { **return** secondLanguage; }

**set** { secondLanguage = value; }

}

**public** **int** Geography

{

**get** { **return** geography; }

**set** { geography = value; }

}

**public** **int** History

{

**get** { **return** history; }

**set** { history = value; }

}

**public** **int** Physics

{

**get** { **return** physics; }

**set** { physics = value; }

}

**public** **int** Chemistry

{

**get** { **return** chemistry; }

**set** { chemistry = value; }

}

**public** **int** Biology

{

**get** { **return** biology; }

**set** { biology = value; }

}

**public** **virtual** **int** getTotalMarks()

{

**this**.totalMarks = maths + english + geography + history + physics + chemistry + biology;

**return** totalMarks;

}

}

}

**Step 4.4.6:** 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.\_10

{

**class** Program

{

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

{

runApp();

}

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

{

ElementarySchoolGradeCard elementary = **new** ElementarySchoolGradeCard ();

elementary.Maths = 90;

elementary.English = 78;

elementary.SecondLanguage = 80;

elementary.SocialScience = 67;

Console.WriteLine("total marks =" +

elementary.getTotalMarks());

MiddleSchoolGradeCard middle = **new** MiddleSchoolGradeCard ();

middle.Maths = 90;

middle.English = 78;

middle.SecondLanguage = 80;

middle.Geography = 87;

middle.History = 76;

Console.WriteLine("total marks =" + middle.getTotalMarks());

HighSchoolGradeCard high = **new** HighSchoolGradeCard ();

high.Maths = 90;

high.English = 78;

high.SecondLanguage = 80;

high.Geography = 87;

high.History = 76;

high.Physics = 90;

high.Chemistry = 76;

high.Biology = 70;

Console.WriteLine("total marks =" + high.getTotalMarks());

}

}

}

**Step 4.4.7:** Building the project

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

**Step 4.4.8:** Publishing and running the project

* From the top menu, select **Debug->Start Without Debugging.**
* This will execute the program in a console window.

**Step 4.4.9:** 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