2.3 Handling Dates 



This section will guide you to:

* Create a Windows Console project in Visual Studio to show date handling
* Create a method doApp() that will demonstrate the use of dates

**Development Environment**

* Visual Studio 2019 Community Version

This guide has five subsections, namely:

* + 1. Creating a Windows Console project in Visual Studio to show date handling
    2. Adding a method doApp() in Program class that will demonstrate the use of dates
    3. Building the project
    4. Publishing and running the project
    5. Pushing the code to your GitHub repositories

**Step** **2.3.1:** Creating a Windows Console project in Visual Studio to show the control flow

* 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 **Phase1Section3.8** and click on **Create.**
* This will create the files for a Windows Console project.

**Step** **2.3.2:** Adding a method doApp() in Program class that will demonstrate the use of dates

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

**using** System;

**namespace** Phase1Section3.\_8

{

**class** Program

{

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

{

doApp();

}

**public** **static** **void** doApp()

{

DateTime dt = DateTime.Now;

Console.WriteLine("Current date and time :" + dt.ToString());

Console.WriteLine("Formatted short date:" + dt.ToShortDateString());

Console.WriteLine("Formatted long date:" + dt.ToLongDateString());

Console.WriteLine("Formatted custom date strings:");

Console.WriteLine(dt.ToString("d"));

Console.WriteLine(dt.ToString("D"));

Console.WriteLine(dt.ToString("F"));

Console.WriteLine(dt.ToString("y"));

Console.WriteLine("Adding 5 days to today:" + dt.AddDays(5).ToShortDateString());

DateTime date1 = **new** DateTime(2019, 1, 2, 4, 0, 15);

DateTime date2 = **new** DateTime(2019, 1, 2, 14, 0, 15);

TimeSpan value = date1.Subtract(date2);

Console.WriteLine("TimeSpan between two dates is {0} hours", value);

}

}

}

**Step 2.3.3:** Building the project

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

**Step 2.3.4:** Publishing and running the project

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

**Step** **2.3.5:** 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