



Islington college
(इस्लिङ्टन कलेज)

Code & Module Title

CS6004NI Application Development

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2018-19 Autumn

Student Name: Rajat Shrestha

London Met ID: 17030954

College ID: np01cp4a170021

Assignment Due Date: 17th January 2020

Assignment Submission Date: 17th January 2020

Word Count: 3000 words

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of Contents

1. Introduction.....	3
1.1. Instructions:.....	3
1.1.1. Adding New Student:.....	4
1.1.2. Sort by name and Registration date:	5
1.1.3. Edit or Delete Row:	6
1.1.4. Viewing the Total enrolled students' chart:	7
1.1.5. Viewing the Weekly report from a certain date:	8
2. Software Architecture	9
2.1. Classes from other sources	10
3. Classes and Methods:.....	11
3.1. Class Diagram	11
3.2. Attributes of Student:	12
3.4. Method Descriptions:	13
3.4.1. Method description of Student Class	13
3.4.2. Method description of StudentForms class.....	14
3.4.3. Method description of Utility Class	16
3.4.4. Method description of Static Program class.....	16
4. Report generation.....	17
5. Core Mechanisms.....	18
5.1. Data Structures:.....	18
5.1.1. List	18
5.2. Insertion Sort:.....	19
6. Reflection	20
7. References	21

Table of Figures:

Figure 1: Main Window with all the interactable components	3
Figure 2: Filling in the Form	4
Figure 3: Added student at the bottom of the list.....	4
Figure 4: Names Sorted in ascending order	5
Figure 5: Names Sorted in descending order (repressing Sort Name button)	5
Figure 6: Updating the Student Details.....	6
Figure 7: Updated Student details	6
Figure 8: Total Students Enrolled Chart Page.....	7
Figure 9: Weekly report form the selected date	8
Figure 10: Software Architecture	9
Figure 11: Overall Class Diagram	11
Figure 12: Flow-Chart for Report Generation	17
Figure 13: Insertion Sort Steps (w3resource, 2019).....	19

Table of Tables:

Table 1: Description of attributes of Student Class	12
Table 2: Method Description of Student Class	13
Table 3: Method Description of StudentForm Class.....	14
Table 4: Method Description of Utility Class.....	16
Table 5: Method Description of Static Program Class.....	16

1. Introduction

This is the report for the first coursework of the Application development module. This document has the detailed information on the student registration management software which allows users to add, edit and view student details. This application describes and gives instructions of all the features to show the reports on total amount of students enrolled in each course (total/weekly). The major features of this program are:

- Stores the Details of the students
- To display the student records in the table right after the student is enrolled.
- To edit the information of already enrolled students.
- To delete the record of enrolled student.
- To display the weekly enrolled number of students in the table format based on chosen date.
- To display the number of students enrolled in each course in bar chart.
- To display the enrollment status of the student.

1.1. Instructions:

The Program when run opens the following window:

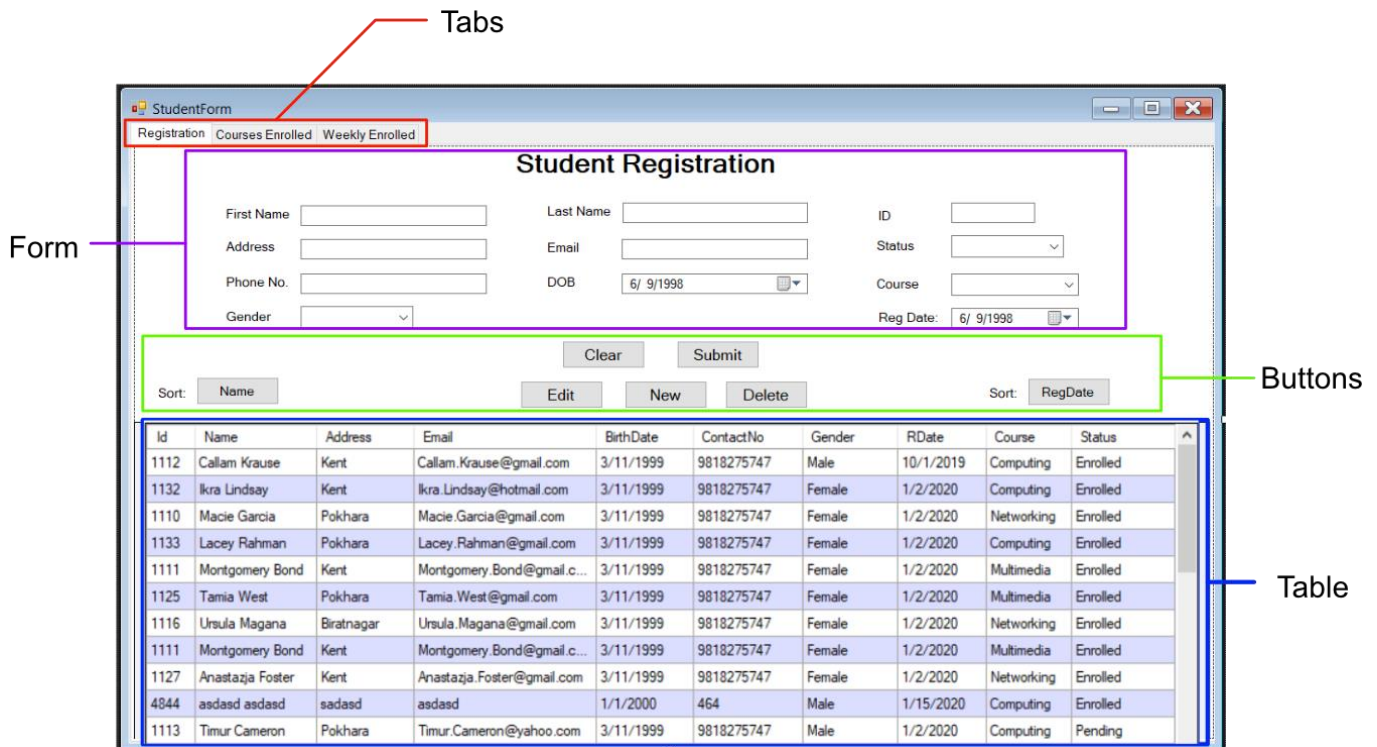


Figure 1: Main Window with all the interactable components

1.1.1. Adding New Student:

1. Fill the Form on the main window (Registration Tab)
2. Click Submit

Student Registration

First Name: Last Name: ID:

Address: Email: Status:

Phone No.: DOB: Course:

Gender: Reg Date:

Sort:

Id	Name	Address	Email	BirthDate	ContactNo	Gender	RDate	Course	Status
1112	Callan Krause	Kent	Callan.Krause@gmail.com	3/11/1999	9818275747	Male	10/1/2019	Computing	Enrolled
1132	Ikra Lindsay	Kent	Ikra.Lindsay@hotmail.com	3/11/1999	9818275747	Female	1/2/2020	Computing	Enrolled
1110	Macie Garcia	Pokhara	Macie.Garcia@gmail.com	3/11/1999	9818275747	Female	1/2/2020	Networking	Enrolled
1133	Lacey Rahman	Pokhara	Lacey.Rahman@gmail.com	3/11/1999	9818275747	Female	1/2/2020	Computing	Enrolled
1111	Montgomery Bond	Kent	Montgomery.Bond@gmail.c...	3/11/1999	9818275747	Female	1/2/2020	Multimedia	Enrolled
1125	Tamia West	Pokhara	Tamia.West@gmail.com	3/11/1999	9818275747	Female	1/2/2020	Multimedia	Enrolled
1116	Ursula Magana	Biratnagar	Ursula.Magana@gmail.com	3/11/1999	9818275747	Female	1/2/2020	Networking	Enrolled
1111	Montgomery Bond	Kent	Montgomery.Bond@gmail.c...	3/11/1999	9818275747	Female	1/2/2020	Multimedia	Enrolled
1127	Anastazja Foster	Kent	Anastazja.Foster@gmail.com	3/11/1999	9818275747	Female	1/2/2020	Networking	Enrolled
4844	asdasd asdasd	sadasd	asdasd	1/1/2000	464	Male	1/15/2020	Computing	Enrolled
1113	Timur Cameron	Pokhara	Timur.Cameron@yahoo.com	3/11/1999	9818275747	Male	1/2/2020	Computing	Pending

Figure 2: Filling in the Form

Student Registration

First Name: Last Name: ID:

Address: Email: Status:

Phone No.: DOB: Course:

Gender: Reg Date:

Sort:

Id	Name	Address	Email	BirthDate	ContactNo	Gender	RDate	Course	Status
1114	Selina Hulme	Biratnagar	Selina.Hulme@yahoo.com	3/11/1999	9818275747	Female	1/7/2020	Networking	Pending
1126	Talitha Galindo	Pokhara	Talitha.Galindo@gmail.com	3/11/1999	9818275747	Female	12/30/2019	Multimedia	Pending
1118	Tyron Blackmore	Kent	Tyron.Blackmore@gmail.com	3/11/1999	9818275747	Male	11/11/2020	Computing	Pending
1138	Rose Yu	Biratnagar	Rose.Yu@gmail.com	3/11/1999	9818275747	Female	1/2/2020	Networking	Pending
1122	Shannan Cherry	Kent	Shannan.Cherry@gmail.com	3/11/1999	9818275747	Male	12/30/2019	Networking	Pending
1115	Kiki Howarth	Pokhara	Kiki.Howarth@yahoo.com	3/11/1999	9818275747	Male	1/29/2020	Computing	Pending
4599	Rajat Shrestha	Samakhushi	rajat.shrestha@eydean.com	3/11/1999	9818275747	Male	12/2/2019	Computing	Pending
1128	Kerrie Oakley	Pokhara	Kerrie.Oakley@yahoo.com	3/11/1999	9818275747	Female	1/2/2020	Computing	Pending
6328	Tejan aamasd	asdasd	asdasda@adadsa.com	1/1/2000	123123	Male	1/17/2020	Multimedia	Enrolled
3019	James Doe	Dune	James.doe@jd.com	11/24/1999	9871237321	Male	1/17/2020	Networking	Pending

Figure 3: Added student at the bottom of the list

1.1.2. Sort by name and Registration date:

1. Click the Sort Button(Name or RegDate) for Sorting in Ascending order
2. Press the same button for Descending Order

StudentForm

Registration Courses Enrolled Weekly Enrolled

Student Registration

First Name Last Name ID

Address Email Status

Phone No. DOB Course

Gender Reg Date:

Sort:

Id	Name	Address	Email	BirthDate	ContactNo	Gender	RDate	Course	Status
1127	Anastazia Foster	Kent	Anastazia.Foster@gmail.com	3/11/1999	9818275747	Female	1/2/2020	Networking	Enrolled
1129	Arran Arellano	Pokhara	Arran.Arellano@gmail.com	3/11/1999	9818275747	Male	12/30/2019	Computing	Pending
4844	asdasd asdasd	sadasd	asdasd	1/1/2000	464	Male	1/15/2020	Computing	Enrolled
1119	Becky McKeen	Pokhara	Becky.McKeen@hotmail.com	3/11/1999	9818275747	Male	1/20/2020	Networking	Pending
1112	Callam Krause	Kent	Callam.Krause@gmail.com	3/11/1999	9818275747	Male	10/1/2019	Computing	Enrolled
1137	Dion Cash	Kent	Dion.Cash@gmail.com	3/11/1999	9818275747	Female	1/6/2020	Multimedia	Pending
1132	Ikra Lindsay	Kent	Ikra.Lindsay@hotmail.com	3/11/1999	9818275747	Female	1/2/2020	Computing	Enrolled
3019	James Doe	Dune	James.doe@jd.com	11/24/1999	9871237321	Male	1/17/2020	Networking	Pending
1123	Jia Ashton	Kent	Jia.Ashton@gmail.com	3/11/1999	9818275747	Female	1/7/2020	Networking	Pending
1128	Kemie Oakley	Pokhara	Kemie.Oakley@yahoo.com	3/11/1999	9818275747	Female	1/2/2020	Computing	Pending
1115	Kiki Howarth	Pokhara	Kiki.Howarth@yahoo.com	3/11/1999	9818275747	Male	1/29/2020	Computing	Pending

Figure 4: Names Sorted in ascending order

StudentForm

Registration Courses Enrolled Weekly Enrolled

Student Registration

First Name Last Name ID

Address Email Status

Phone No. DOB Course

Gender Reg Date:

Sort:

Id	Name	Address	Email	BirthDate	ContactNo	Gender	RDate	Course	Status
1116	Ursula Magana	Biratnagar	Ursula.Magana@gmail.com	3/11/1999	9818275747	Female	1/2/2020	Networking	Enrolled
1118	Tyron Blackmore	Kent	Tyron.Blackmore@gmail.com	3/11/1999	9818275747	Male	11/11/2020	Computing	Pending
1113	Timur Cameron	Pokhara	Timur.Cameron@yahoo.com	3/11/1999	9818275747	Male	1/2/2020	Computing	Pending
6328	Tejan aamasd	asdasd	asdasda@adadsa.com	1/1/2000	123123	Male	1/17/2020	Multimedia	Enrolled
1125	Tamia West	Pokhara	Tamia.West@gmail.com	3/11/1999	9818275747	Female	1/2/2020	Multimedia	Enrolled
1126	Talitha Galindo	Pokhara	Talitha.Galindo@gmail.com	3/11/1999	9818275747	Female	12/30/2019	Multimedia	Pending
1122	Shannan Cherry	Kent	Shannan.Cherry@gmail.com	3/11/1999	9818275747	Male	12/30/2019	Networking	Pending
1114	Selina Hulme	Biratnagar	Selina.Hulme@yahoo.com	3/11/1999	9818275747	Female	1/7/2020	Networking	Pending
1138	Rose Yu	Biratnagar	Rose.Yu@gmail.com	3/11/1999	9818275747	Female	1/2/2020	Networking	Pending
4599	Rajat Shrestha	Samakhushi	rajat.shrestha@eydean.com	3/11/1999	9818275747	Male	12/2/2019	Computing	Pending
1135	Raife Brandt	Kent	Raife.Brandt@gmail.com	3/11/1999	9818275747	Female	1/14/2020	Networking	Pending

Figure 5: Names Sorted in descending order (repressing Sort Name button)

1.1.3. Edit or Delete Row:

- 1) Select the row to update
- 2) The information appears on the forms
- 3) Edit the form
- 4) Press Edit Button or to delete press delete
- 5) The content is updated

Student Registration

First Name: Talitha, Last Name: Galindo, ID: 1126, Address: Pokhara, Email: Talitha.Galindo@gmail.com, Status: Pending, Phone No.: 9818275747, DOB: 11 Jun 1996, Course: Multimedia, Gender: Female, Reg Date: 30 Dec 2019

Sort: Name | Edit | New | Delete | Sort: RegDate

Id	Name	Address	Email	BirthDate	ContactNo	Gender	RDate	Course	Status
1126	Talitha Galindo	Pokhara	Talitha.Galindo@gmail.com	3/11/1999	9818275747	Female	12/30/2019	Multimedia	Pending
1118	Tyron Blackmore	Kent	Tyron.Blackmore@gmail.com	3/11/1999	9818275747	Male	11/11/2020	Computing	Pending
1138	Rose Yu	Biratnagar	Rose.Yu@gmail.com	3/11/1999	9818275747	Female	1/2/2020	Networking	Pending
1122	Shannan Chery	Kent	Shannan.Chery@gmail.com	3/11/1999	9818275747	Male	12/30/2019	Networking	Pending
1115	Kiki Howarth	Pokhara	Kiki.Howarth@yahoo.com	3/11/1999	9818275747	Male	1/29/2020	Computing	Pending
4599	Rajat Shrestha	Samakhushi	rajat.shrestha@eydean.com	3/11/1999	9818275747	Male	12/2/2019	Computing	Pending
1128	Kerrie Oakley	Pokhara	Kerrie.Oakley@yahoo.com	3/11/1999	9818275747	Female	1/2/2020	Computing	Pending
6328	Tejan aamasd	asdasd	asdasda@adadsa.com	1/1/2000	123123	Male	1/17/2020	Multimedia	Enrolled
3019	James Doe	Dune	James.doe@jyd.com	11/24/1999	9871237321	Male	1/17/2020	Networking	Pending
1116	Ursula Magana	Biratnagar	Ursula.Magana@gmail.com	3/11/1999	9878879122	Female	1/2/2020	Networking	Pending

Figure 6: Updating the Student Details

Student Registration

First Name: Talitha, Last Name: Galindo, ID: 1126, Address: Pokhara, Email: Talitha.Galindo@gmail.com, Status: Enrolled, Phone No.: 9818275747, DOB: 11 Jun 1996, Course: Multimedia, Gender: Female, Reg Date: 30 Dec 2019

Sort: Name | Edit | New | Delete | Sort: RegDate

Id	Name	Address	Email	BirthDate	ContactNo	Gender	RDate	Course	Status
1118	Tyron Blackmore	Kent	Tyron.Blackmore@gmail.com	3/11/1999	9818275747	Male	11/11/2020	Computing	Pending
1138	Rose Yu	Biratnagar	Rose.Yu@gmail.com	3/11/1999	9818275747	Female	1/2/2020	Networking	Pending
1122	Shannan Chery	Kent	Shannan.Chery@gmail.com	3/11/1999	9818275747	Male	12/30/2019	Networking	Pending
1115	Kiki Howarth	Pokhara	Kiki.Howarth@yahoo.com	3/11/1999	9818275747	Male	1/29/2020	Computing	Pending
4599	Rajat Shrestha	Samakhushi	rajat.shrestha@eydean.com	3/11/1999	9818275747	Male	12/2/2019	Computing	Pending
1128	Kerrie Oakley	Pokhara	Kerrie.Oakley@yahoo.com	3/11/1999	9818275747	Female	1/2/2020	Computing	Pending
6328	Tejan aamasd	asdasd	asdasda@adadsa.com	1/1/2000	123123	Male	1/17/2020	Multimedia	Enrolled
3019	James Doe	Dune	James.doe@jyd.com	11/24/1999	9871237321	Male	1/17/2020	Networking	Pending
1116	Ursula Magana	Biratnagar	Ursula.Magana@gmail.com	3/11/1999	9878879122	Female	1/2/2020	Networking	Pending
1126	Talitha Galindo	Pokhara	Talitha.Galindo@gmail.com	6/11/1996	9818275747	Female	12/30/2019	Multimedia	Enrolled

Figure 7: Updated Student details

1.1.4. Viewing the Total enrolled students' chart:

1. Go to the Second tab (Courses Enrolled)
2. The Chart is generated

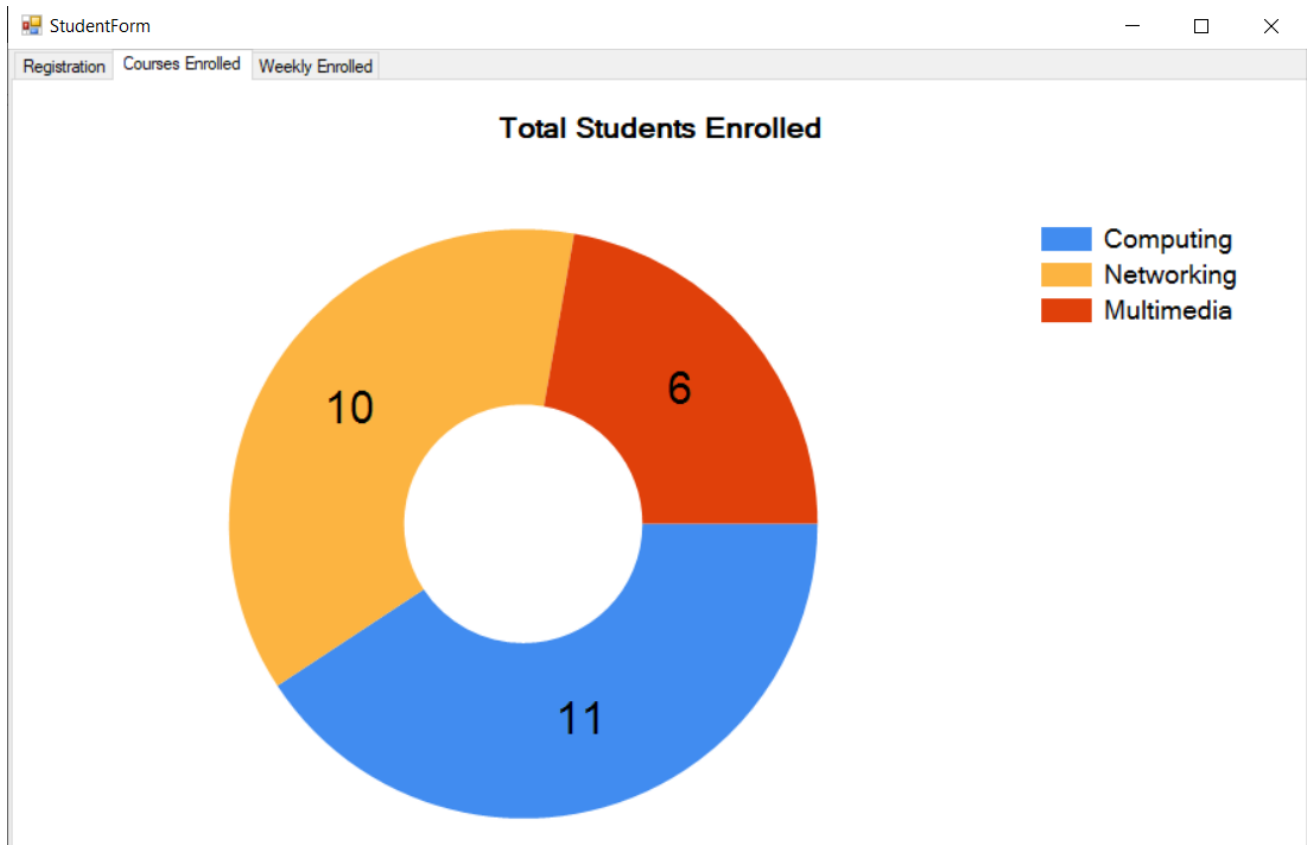


Figure 8: Total Students Enrolled Chart Page

1.1.5. Viewing the Weekly report from a certain date:

1. Go to the Third tab (Weekly Report)
2. Select the day from when weekly report is to be generated
3. The Weekly report from the selected date is shown with the count of students in each course and their information in the table below in tab-3.

Select the date to view the weekly report

Friday, January 3, 2020

January 2020						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Today: 1/17/2020

	Count
Multimedia	1
Networking	2

StudentForm

Registration Courses Enrolled Weekly Enrolled

Select the date to view the weekly report

Friday, January 3, 2020

	Program	Count
▶	Multimedia	1
	Networking	2

Id	Name	Address	Email	BirthDate	ContactNo	Gender	RDate	Course	Status
1137	Dion Cash	Kent	Dion.Cash@gmail.com	3/11/1999	9818275747	Female	1/6/2020	Multimedia	Pending
1123	Jia Ashton	Kent	Jia.Ashton@gmail.com	3/11/1999	9818275747	Female	1/7/2020	Networking	Pending
1114	Selina Hulme	Biratnagar	Selina.Hulme@yahoo.com	3/11/1999	9818275747	Female	1/7/2020	Networking	Pending

Figure 9: Weekly report form the selected date

2. Software Architecture

This software is quite simple with only 4 classes built in the CourseWork namespace. The classes import essential system components. The following diagram represents how the overall system function, and the how the system utilized imported modules:

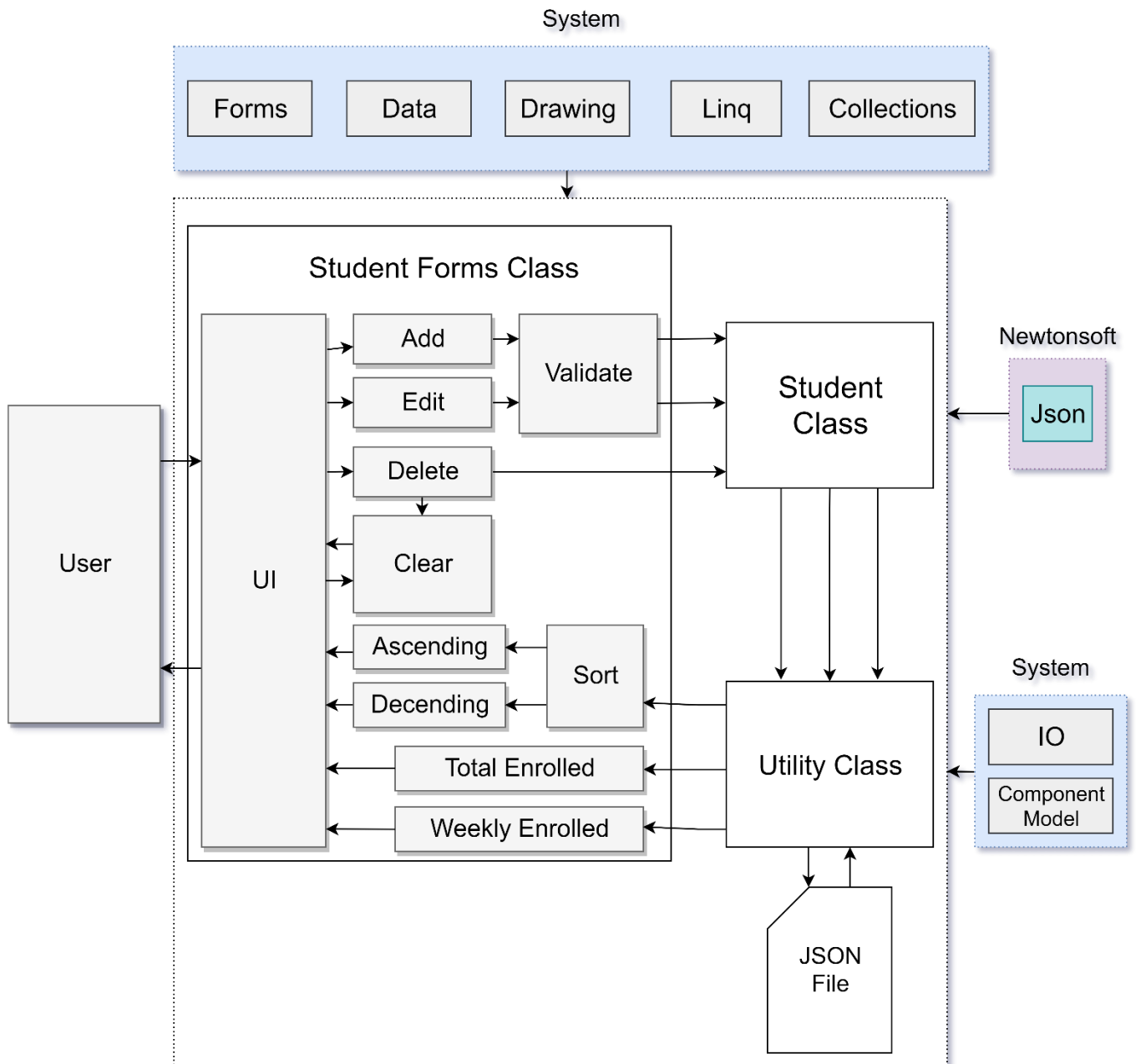


Figure 10: Software Architecture

(The colored Modules are imported i.e. System and Newtonsoft)

2.1. Classes from other sources

Various classes were used in the construction of the Form to make the development easier. The Student form class was directly extended from the Form Class; Most of the System classes were used for storing data in objects, read/write purposes and for basic functionalities; The Newtonsoft.Json package was installed for helping in serialization on the Student class.

Windows.Forms (Form)

This namespace is used to inherit the form class used in the StudentForm class. Its main purpose is to provide an easy GUI interface for creating form. Windows Forms Applications are the applications which are developed by using Windows Forms (WinForms) and are supported on windows. WinForms limited to Windows Forms Applications and provides different type of UI elements (Microsoft, 2019). In this project form is expanded to be used as an application for student registration and report preview.

Other used classes and packages:

These classes were used in various parts of the project which are provided by the system.

- Newtonsoft.Json (JsonConvert)
- System.Collections.Generic (List<T>)
- System.Data (DataTable)
- System.Drawing (Chart)
- System.IO (StreamWriter and StreamReader)
- System.Linq
- System.ComponentModel

(Note: The Newtonsoft.Json package requires to be installed manually.)

3. Classes and Methods:

3.1. Class Diagram

Class diagram is fundamental diagram type in UML, which is used to represent the static relationship between different components of an application, which is key to understand the overall working of any system. A class is group of objects with common state and behavior and is represented in a UML class diagram by putting Name of the class at the first box, attributes of the class at the middle box and finally the methods of that class at the last box (Pilone & Pitman, 2005).

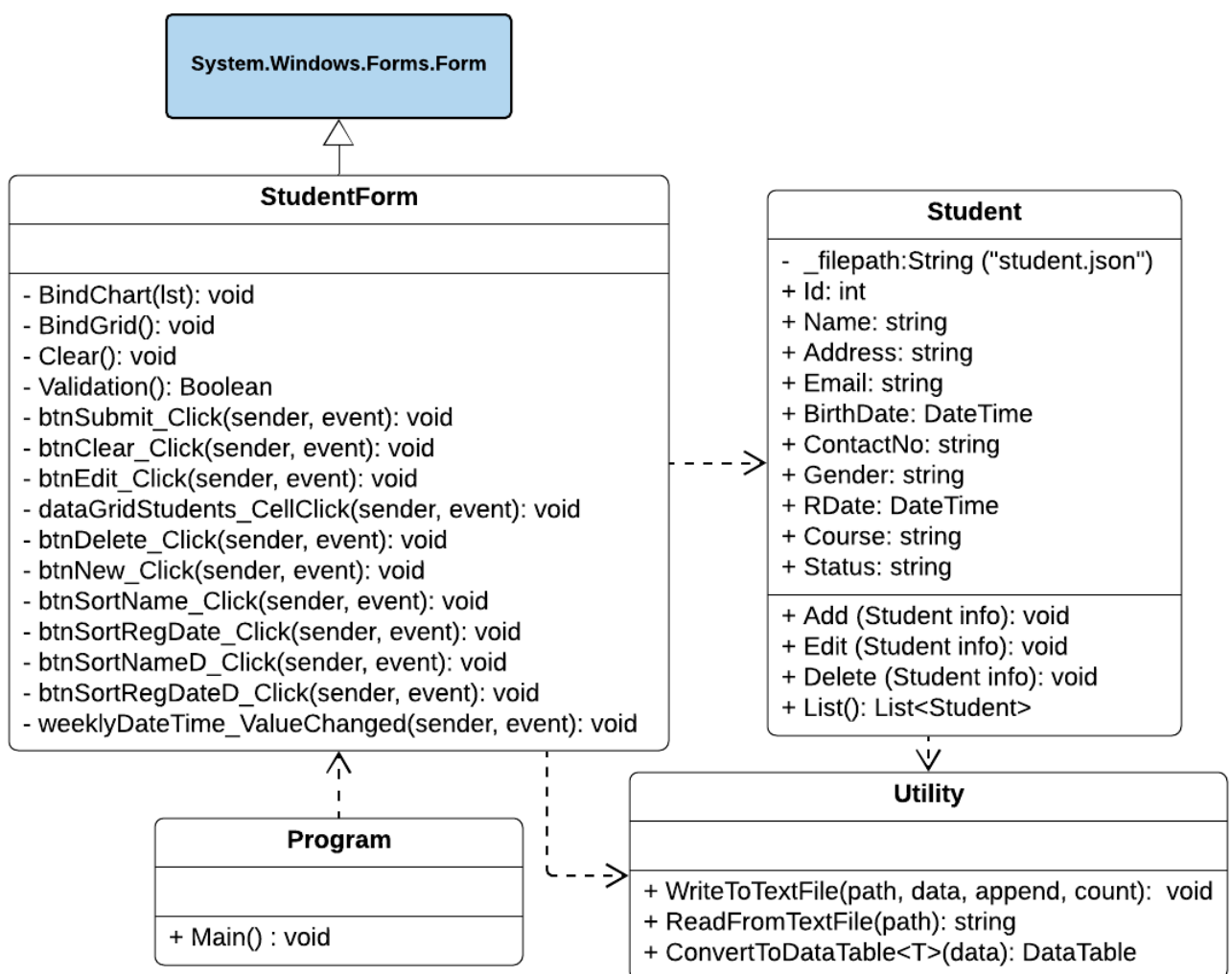


Figure 11: Overall Class Diagram

3.2. Attributes of Student:

The above class diagram shows the attributes used in the Student class which is used to define a Student Object. The attributes and their significance in detail is as follows:

Table 1: Description of attributes of Student Class

Attribute	Description
_filepath	Name of the json file to store the Student information in String.
Id	Unique ID in int auto generated during the creation.
Name	Name of the student consisting of First and Last Name in String.
Address	Address of the Student in String
Email	E-mail address of the student in String
Birthdate	Date of Birth of the Student in DateTime format.
ContactNo	Phone Number of the Student in String.
Gender	Gender of the Student in String.
RDate	Registration Date of the Student.
Course	Enrolled Course of the Student in String.
Status	Current Enrolment Status of the Student

(Note: The other Classes do not have any key attributes, so they are skipped.)

3.4. Method Descriptions:

A method description contains brief description of statements that are put together to perform an operation in a class (Jenkov, 2015). It helps to make the code modular and reuse the code. The C# program to manage the information regarding students were created with four different classes within the “CourseWork” namespace.

3.4.1. Method description of Student Class

Table 2: Method Description of Student Class

Modifiers, Return Type and Method	Method description
public void Add (Student info)	The add method is responsible for adding student information received from the from input to the text file. This method also gives random id to each enrolled student.
public void Edit (Student info)	The edit option to update details of enrolled students.
public void Delete (Student info)	The delete method is to delete the records of the already enrolled student. If the user wants to delete records of student.
public List < Student > List ()	This method reads data from text file and shows it on the data grid table.

3.4.2. Method description of StudentForms class

Table 3: Method Description of StudentForm Class

Modifiers, Return Type and Method	Method description
public StudentForm()	Method which initializes all the components and sets up all the variables required for the components
private Boolean Validation()	A method which checks all the entry fields and returns a Boolean to indicate its validation.
private void BindChart(List<Student> list)	The function of this method is to add the Student enrolment details in each subject in the chart. The Bind Chart is used to show the count of each course on chart in form of a bar graph.
private void BindGrid()	The method which converts the list of students to a DataTable so it can be displayed in the table. This method also initiates the BindChart() method which displays the enrolment chart.
private void Clear()	The method in StudentForm table is to clear all the fields in the Form from where all the information of the students is entered.
private void btnSubmit_Click(object sender, EventArgs e)	This method is for adding new student details from the forms. If the validation returns true, all the input information from the form fields is stored in the JSON file and is displayed on the table.
private void btnClear_Click(object sender, EventArgs e)	This method invokes Clear function which clears the inputs fields
private void btnEdit_Click(object sender, EventArgs e)	This method calls the edit method in student class to edit the student records updating the student details whenever user clicks the Edit button.
private void dataGridStudents_CellClick(object sender, DataGridViewCellEventArgs e)	This method returns the content of the selected row in the DataGrid table for editing or deleting.

private void btnDelete_Click (object sender, EventArgs e)	The function of this method is this method is to call delete method to delete the record of selected student. Before deleting the program asks for confirmation.
private void btnNew_Click (object sender, EventArgs e)	The method got activated when the user clicks on New button and then it allows to enter the new record of students.
private void btnSortName_Click (object sender, EventArgs e)	The method invoked after user presses the sort name option which uses insertion sort to sort all the names in ascending order.
private void btnSortRegDate_Click (object sender, EventArgs e)	Similarly, this method sorts the DataTable according to the registration date in ascending order. The button for descending order is
private void btnSortNameD_Click (object sender, EventArgs e)	The method to arrange the names in the DataTable descending order.
private void btnSortRegDateD_Click (object sender, EventArgs e)	The method to arrange the Registration date in the DataTable descending order.
private void weeklyDateTime_ValueChanged (object sender, EventArgs e)	This method is invoked after the user selects a date in the weekly report calendar. It then queries all the stored data and views only the students who have been enrolled after 7 days of the selected date.

3.4.3. Method description of Utility Class

Table 4: Method Description of Utility Class

Modifiers, Return Type and Methods	Method description
public static void WriteToFile (string path, string data, bool append = true, int count = 1)	Method to store the information received from the form in the text file. It uses StreamWriter to write the JSON structured data in the file.
public static string ReadFromFile (string path)	Method to read the information of students from the text file. The method first checks either or not file exists. If the file exists retrieves the information inside it by using StreamReader to read JSON file. This method returns the read data in string format.
public static DataTable ConvertToDataTable <T>(IList<T> data)	This method converts the data and returns it in the DataTable format so that the information of the students can be displayed on the table.

3.4.4. Method description of Static Program class

Table 5: Method Description of Static Program Class

Modifiers, Return Type and Method	Method description
static void Main ()	The Static method to initiate a new StudentForm which runs the overall program.

4. Report generation

This application has a feature to generate and view all the students enrolled so far in the application. This section describes how these report generation works by providing a brief description and a descriptive diagram (flow-chart).

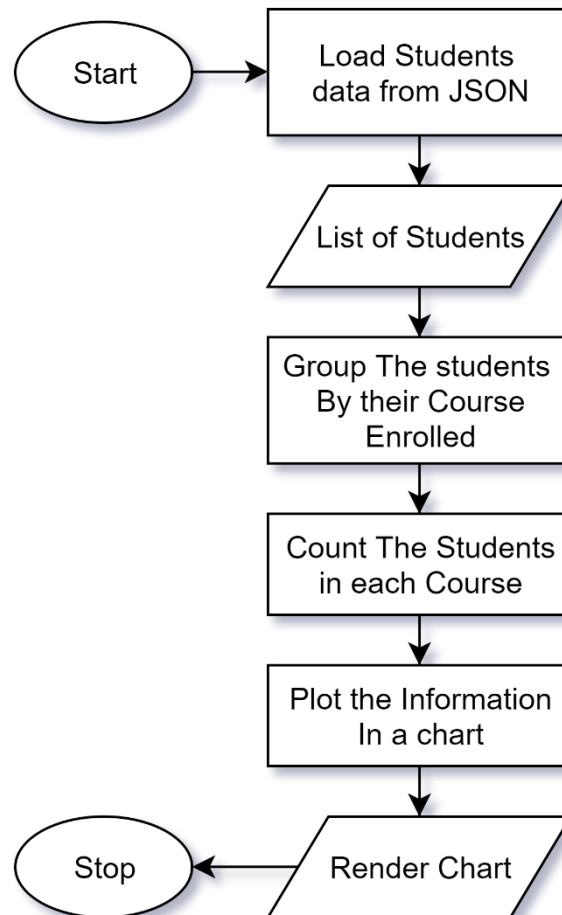


Figure 12: Flow-Chart for Report Generation

5. Core Mechanisms

5.1. Data Structures:

A data structure is a specialized class created for organizing, processing, retrieving and storing data efficiently and can be a key aspect in the performance of any software. There are many simple to complex data structures which are used for various purposes and are arranged to work effectively in any software. Each data structure contains information about the data values, relationships between the data and functions that can be applied to the data (Storer, 2012). Some examples of data structures are: Arrays, Stacks, Queues, LinkedList, trees. Among all the available data structures, list data structure is used throughout the projects for storing data and for various operations.

5.1.1. List

In C#, List is a generic collection which is used to store the elements or objects in the form of a list, and it is defined under System.Collections.Generic namespace. It provides the same functionality as ArrayList, but a list is a generic whereas ArrayList is a non-generic collection which means the size of a list is dynamic and can grow gradually as needed (MSDN, 2019). A List class has 3 constructors which are used to create a list and the constructors are as follows:

- List<T>(): For initiating empty list
- List<T>(Enumerable): List with specified object type.
- List<T>(Int32): For creating a list with fixed capacity

Example:

```
List<string> my_list1 = new List<string>() { "Kathmandu", "See Sharp" };
```

5.2. Insertion Sort:

Insertion sort is a simple linear sort based on the idea that one element from the input elements is consumed in each iteration to find its correct position. It iterates the input elements one by one by growing the sorted array at each iteration then repeatedly compares the current element with the largest value in the sorted array. If the current element is greater, the element is kept on its place, else it finds its correct position in the sorted array and moves it to that position. Finally, the sort is complete after all the elements are correctly sorted. Insertion sort has the time complexity of $O(n^2)$ which is not the best, but it works well on small to medium projects and has lesser memory consumption than the other more efficient sorts like Merge sort (Mahmoud, 2011). The following are some of the important characteristics of Insertion Sort:

- Efficient for smaller data sets and partially sorted instances, but very inefficient for large lists.
- It is better than Selection Sort and Bubble Sort algorithms but worse than Merge Sort.
- Its space complexity is less requiring only a single additional memory space.
- It is stable sorting technique, as it does not change the relative order of elements that are equal.

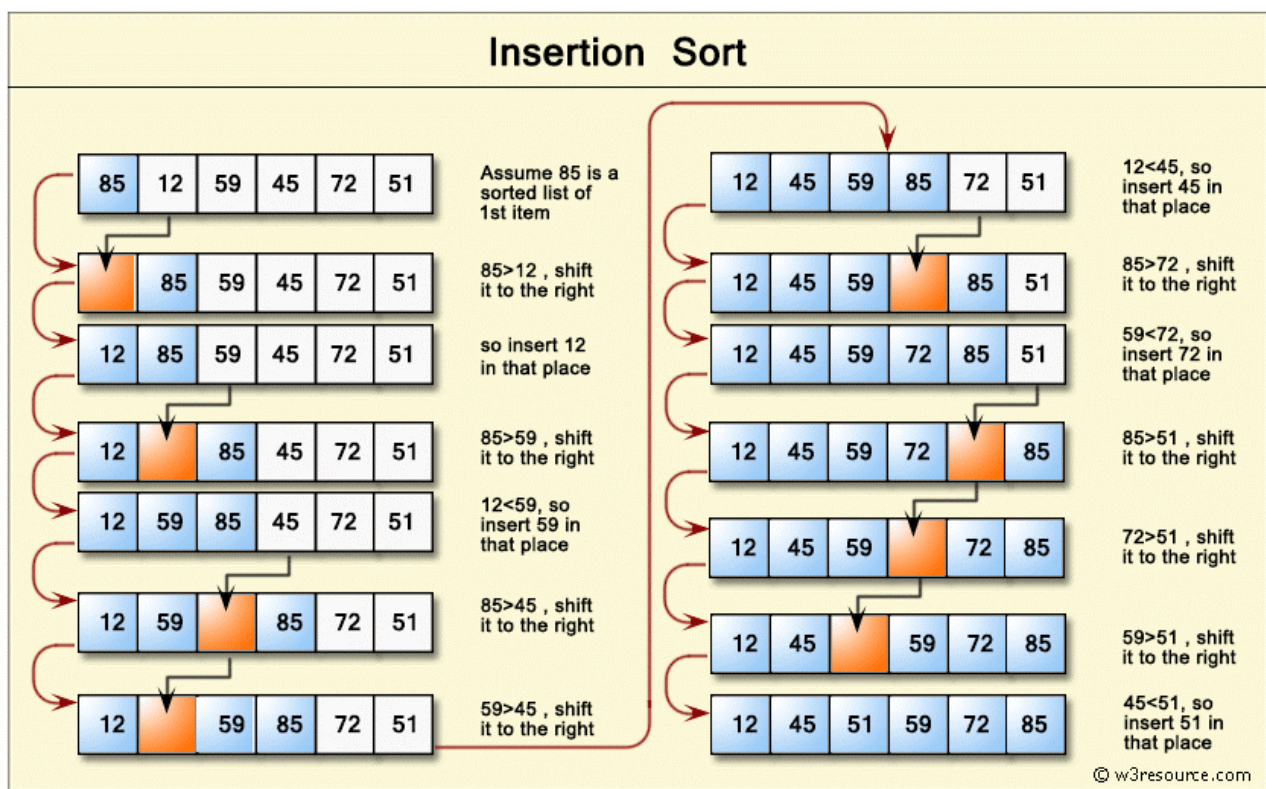


Figure 13: Insertion Sort Steps (w3resource, 2019)

6. Reflection

There Was little to no problems faced while developing the windows form in C# using Visual Studio IDE. C# is a modern, simple, object-oriented, and type-safe programming language identical to C++, C and Java (MSDN, 2019). Since, most of the syntax for programming in C# was identical to syntax in Java there was little to no problem and a lot of productivity while coding in C#. The main features and advantages along with the disadvantages are as follows: of using C# with Visual Studio IDE are as follows:

Advantages:

1. C# being Object Oriented Programming Language,
2. C# has seamless capability to integrate with Windows,
3. C# is a compiled language,
4. Visual Studio has clean UI and good auto code refactoring for C# development,
5. Easy development due to intuitive design of Visual Studio,
6. Visual Studio has good debugging and other important development tools for C#

Disadvantages:

1. Visual Studio is quite Memory Intensive and takes a lot of RAM while running,
2. Some of the features such as hotkey for commenting code and Find/Replace is different in Visual Studio than in other Text Editors and IDEs,
3. Visual Studio is not open source,
4. C# does not work on other platforms (e.g. Linux).

Therefore, C# is a good programming language with a lot of advantages outweighing its disadvantages which allows developers to build powerful windows application easily along with Visual Studio IDE. C# and Visual Studio was quite useful in development of the given project which might have taken more time in other programming language.

7. References

Hopping, C., 2018. *What is a graphical user interface?*. [Online]

Available at: <https://www.itpro.co.uk/operating-systems/30248/what-is-a-graphical-user-interface>

[Accessed 18 1 2019].

Jacobson, N. & Thornton, A., 2004. It is Time to Emphasize ArrayLists over Arrays. *inroads – The SIGCSE Bulletin*, 36(4), pp. 88-92.

Jenkov, J., 2015. *Java Methods*. [Online]

Available at: <http://tutorials.jenkov.com/java/methods.html>

[Accessed 17 Jan 2018].

Mahmoud, H. M., 2011. *Sorting: A Distribution Theory*. First ed. New Jersey: John Wiley & Sons.

Microsoft, 2019. *Form Class*. [Online]

Available at: <https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.form?view=netframework-4.8>

[Accessed 16 January 2020].

MSDN, 2019. *A tour of the C# language*. [Online]

Available at: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>

[Accessed 16 January 2020].

MSDN, 2019. *List<T> Class*. [Online]

Available at: <https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1?view=netframework-4.8>

[Accessed 16 January 2020].

Pilone, D. & Pitman, N., 2005. *UML 2.0 in a Nutshell*. First ed. Sebastapool: O'Reilly Media.

Rouse, M., 2014. *What is object-oriented programming (OOP)? - Definition from WhatIs.com*. [Online]

Available at: <http://searchmicroservices.techtarget.com/definition/object-oriented-programming-OOP>

[Accessed 18 Jan 2018].

Storer, J. A., 2012. *An Introduction to Data Structures and Algorithms*. First ed. Berlin: Springer Science & Business Media.

w3resource, 2019. *Python Data Structures and Algorithms: Insertion sort*. [Online]

Available at: <https://www.w3resource.com/python-exercises/data-structures-and-algorithms/python-search-and-sorting-exercise-6.php>

[Accessed 16 January 2020].

