<<Key Language Features Description>>

1- Classes: User defined data type with functions

2- List<T>: A collection of strongly typed objects that can be accessed by index and having methods for manipulation.

3- Constructor: A function that gets called when an object is created.

4- Variables: A storage area that is used to store values of various data types.

5- Output Method: Writes the specified data, to the standard output stream.

6- While loop: loops through a block of code as long as a specified condition is true.

7- Input Method: Reads the next line of characters from the standard input stream.

8- Labels: A checkpoint you can jump to using goto.

9- Goto: Transfer the program to the label statement.

10- Exception Handling: Provides a way to transfer control form one part of a program to another.

11- Increment: Increases a digit variable by 1.

12- Type Conversion: Changes a variable's data type.

13- Break: Used to get out of a statement.

14- If(Condition) else: If condition holds true, program executes statement after if, if condition is false program executes statement after else.

15-   User Defined Methods: Method does what the program wants.

16-   Continue: Skips one iteration.

17-   List Manipulation: Adding elements to the list or removing them.

18-   String Formatting: Sets coordinates for the text to appear in.

19-   Max(): Gets the maximum value in a group of values.

20-   Min(): Get the minimum value in a group of values.

21-   Objects: An entity that has a state and behavior.

<<Language Features map>>

Language Feature	Class/Filename	Line No.
Class	bookClub/Program.cs	6
List<T>	bookClub/Program.cs	8,.....
Constructors	bookClub/Program.cs	14
Variables	bookClub/Program.cs	16,....
Output Method	bookClub/Program.cs	18,....
While loop	bookClub/Program.cs	19,....
Input Method	bookClub/Program.cs	23,...
Labels	bookClub/Program.cs	26
goto	bookClub/Program.cs	35,...
Exception Handling	bookClub/Program.cs	27-36,.....
Increment	bookClub/Program.cs	40,...
Typer Conversion	bookClub/Program.cs	51,...
Break	bookClub/Program.cs	76,...
If(condition) else	bookClub/Program.cs	95,...
User defined Method	bookClub/Program.cs	110,...
for (initialize;condition;increment)	bookClub/Program.cs	114,...
Continue	bookClub/Program.cs	117,...
List Manipulation	bookClub/Program.cs	24, 136, ...
String Formatting	bookClub/Program.cs	158,...
Max(), Min()	bookClub/Program.cs	186 and 189,....
Objects	Program/Program.cs	231

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
namespace ConsoleApp1
{
    public class bookClub
    {
        public List<string> Members = new List<string>();
        public List<int> Phone = new List<int>();
        public List<string> Email = new List<string>();
        public List<string> Books = new List<string>();
        public List<int> Pages = new List<int>();
        public List<string> Category = new List<string>();
        public bookClub()
        {
            int counter = 1, phone_pageNum; //Can't use readline to initilize a list, so read a subsitiute variable then add it to the list instead
            string names;
            Console.WriteLine("Creating a new group of at least 10 members\n");
            while (true)
            {
                int innerCounter = 1;
                Console.Write("Enter the name of the memeber: ");
                names = Console.ReadLine();
                Members.Add(names);
                Console.Write("Enter the mobile number of the memeber: ");
            checkpoint:
                try //Exception handling to prevent crashes in running the code
                {
                    phone_pageNum = Convert.ToInt32(Console.ReadLine());
                    Phone.Add(phone_pageNum);
                }
                catch(Exception a)
                {
                    Console.WriteLine(a.Message + "\nTry entering again");
                    goto checkpoint;
                }
                Console.Write("Enter the email of the memeber: ");
                names = Console.ReadLine();
                Email.Add(names);
                counter++;
                while (true)
                {

                    Console.Write("\nEnter a book read by the member: ");
                    names = Console.ReadLine();
                    Books.Add(names);
```

```csharp
                    Console.Write("\nEnter number of pages of the book: ");
                checkpoint2:
                    try
                    {
                        phone_pageNum = Convert.ToInt32(Console.ReadLine());
                        Pages.Add(phone_pageNum);
                    }
                    catch(Exception a)
                    {
                        Console.WriteLine(a.Message + "\nTry entering again");
                        goto checkpoint2;
                    }
                    Console.Write("\nEnter the category of the book: ");
                    names = Console.ReadLine();
                    Category.Add(names);
                    innerCounter++;
                    if (innerCounter > 5)
                    {
                        char choice;
                    label1:
                        Console.WriteLine("Do you wish to add another book? (Type Y to add, Type N to exit)");
                        try { choice = Console.ReadLine()[0]; }
                        catch (Exception x) { Console.WriteLine( "Wrong input. Try again\n"); goto label1;}
                        if (choice == 'n' || choice == 'N')
                        {
                            Console.WriteLine("Books initilizing done.\n");
                            Books.Add("-");//Key character separating books for each memeber. For instance if I want to reach the books for the second memeber I'll look for the books between the first and second dash
                            Pages.Add(-1);
                            Category.Add("-");
                            break;
                        }
                        else if (choice == 'y' || choice == 'Y')
                        {

                        }
                        else
                        {
                            Console.WriteLine("Wrong input, go again");
                            goto label1;
                        }
                    }
                }
            }
            if (counter > 10)
            {
                char choice;
            label2:
                Console.WriteLine("Do you wish to add another memeber? (Type Y to add, Type N to exit)");
                choice = Console.ReadLine()[0];
                if (choice == 'n' || choice == 'N')
                {
                    Console.WriteLine("Members initilizing done, now exiting.\n");
```

```csharp
                        Console.WriteLine("Members initilizing done, now exiting.\n");
                        break;
                    }
                    else if (choice == 'y' || choice == 'Y')
                    { }
                    else
                    {
                        Console.WriteLine("Wrong input, go again");
                        goto label2;
                    }
                }
            }
        }
        public void groupStats()//Number of books read by the whole group members. Number of pages read by the whole group members
        {
            int pageSum = 0, bookSum = 0;
            Console.WriteLine("Number of books read by the group: ");
            for (int i = 0; i < Books.Count; i++)
            {
                if (Books[i] == "-")
                    continue;
                bookSum++;
            }
            Console.WriteLine(bookSum + "\n");
            Console.WriteLine("Number of Pages read by the group: ");
            for (int i = 0; i < Pages.Count; i++)
            {
                if (Pages[i] != -1)
                    pageSum += Pages[i];
            }
            Console.WriteLine(pageSum + "\n");
        }
        public void categoryRanks() //Ranking of books categories mostly read by the group members [show the
                                    //number alongside the category in the ranked list].
        {
            List<string> categoryTemp = new List<string>();
            for (int i = 0; i < Category.Count; i++)//Adding the categories to a substitute list
                categoryTemp.Add(Category[i]);
            while (categoryTemp.Contains("-"))
            {
                categoryTemp.Remove("-");
            }
            Console.WriteLine("<<Categories Ranks By Most Read>>");
            Console.WriteLine("Category\t\t|\t  Frequency");
            Console.WriteLine("---------------------------------------------------------------");
            while (categoryTemp.Count != 0)//Sort by most frequent
            {
                int countFreq = 0;
                string freqElement = "";
                for (int i = 0; i < categoryTemp.Count; i++)
                {
                    string tempFrequentElement = categoryTemp[i];
                    int tempFreqCount = 0;
                    for (int j = 0; j < categoryTemp.Count; j++)
                        if (categoryTemp[j] == tempFrequentElement)
                            tempFreqCount++;
                    if (tempFreqCount > countFreq)
                    {
                        freqElement = tempFrequentElement;
                        countFreq = tempFreqCount;
                    }
                }
                Console.WriteLine("{0,-24}|{1,14}", freqElement, countFreq);

                while (categoryTemp.Contains(freqElement))
                {
                    categoryTemp.Remove(freqElement);
                }
            }
        }
        public void membersRanksBasedOnBooksRead() //Ranking of group members based on number of books read[show the number
                                                   //alongside the member's name in the ranked list].
        {
            List<int> numOfBooks = new List<int>();
            int booksForEachMember = 0;
            for (int i = 0; i < Books.Count; i++)
            {
                if (Books[i] == "-")
                {
                    numOfBooks.Add(booksForEachMember);
                    booksForEachMember = -1;
                }
                booksForEachMember++;
            }
            Console.WriteLine("\n<<Members Ranks Based on Most Books>>");
            Console.WriteLine("Memeber\t\t\t|\tNumber Of Books");
            Console.WriteLine("---------------------------------------------------------------");
            for (int j = 0; j < Members.Count; j++)
            {
                for (int i = 0; i < numOfBooks.Count; i++)
                {
                    if (numOfBooks[i] == numOfBooks.Max())
                    {
                        Console.WriteLine("{0, -24}|{1,14}", Members[i], numOfBooks[i]);
```

```csharp
                                    numOfBooks[i] = numOfBooks.Min() - 1;
                                    break;
                                }
                            }
                        }
                    }
        public void memebersRanksBasedOnPages() //Ranking of group members based on number of pages read [show the number
                    {                                       //alongside the member's name in the ranked list].
                        List<int> numOfPages = new List<int>();
                        int pagesForEachMember = 0;
                        for (int i = 0; i < Pages.Count; i++)
                        {
                            if (Pages[i] == -1)
                            {
                                numOfPages.Add(pagesForEachMember);
                                pagesForEachMember = 0;
                                continue;
                            }
                            pagesForEachMember += Pages[i];
                        }
                        Console.WriteLine("\n<<Members Ranks Based on Most Pages read>>");
                        Console.WriteLine("Member\t\t\t|\tNumber of Pages");
                        Console.WriteLine("----------------------------------------------------------------");
                        for (int i = 0; i < Members.Count; i++)
                        {
                            for (int j = 0; j < numOfPages.Count; j++)
                            {
                                if (numOfPages[j] == numOfPages.Max())
                                {
                                    Console.WriteLine("{0, -24}|{1,14}", Members[j], numOfPages[j]);
                                    numOfPages[j] = numOfPages.Min() - 1;
                                    break;
                                }
                            }
                        }

                    }
            }
    class Program
    {
        static void Main(string[] args)
        {
            bookClub Group1 = new bookClub();//Every object of bookClub is a group of readers
            Group1.groupStats();
            Group1.categoryRanks();
            Group1.membersRanksBasedOnBooksRead();
            Group1.memebersRanksBasedOnPages();
        }
    }
}
```

<<Output Screenshots>>

```
Creating a new group of at least 10 members

Enter the name of the memeber: Mohammed Fahad
Enter the mobile number of the memeber: 0511111111
Enter the email of the memeber: a@a.a

Enter a book read by the member: Fevre Dream

Enter number of pages of the book: 350

Enter the category of the book: Horror

Enter a book read by the member: Lord of The Rings

Enter number of pages of the book: 600

Enter the category of the book: Fantasy

Enter a book read by the member: Crime and Punishment

Enter number of pages of the book: 450

Enter the category of the book: Psychological

Enter a book read by the member: The Shining

Enter number of pages of the book: 300

Enter the category of the book: Horror

Enter a book read by the member: Da Vinci Code

Enter number of pages of the book: 600
```

```
Enter the category of the book: Mystery
Do you wish to add another book? (Type Y to add, Type N to exit)
y

Enter a book read by the member: A Song of Ice and Fire

Enter number of pages of the book: 900

Enter the category of the book: Fantasy
Do you wish to add another book? (Type Y to add, Type N to exit)
N
Books initilizing done.

Enter the name of the memeber: Nasser Ali
Enter the mobile number of the memeber: 0522222222
Enter the email of the memeber: b@a.a

Enter a book read by the member: The Lighthouse

Enter number of pages of the book: 500

Enter the category of the book: Horror

Enter a book read by the member: Fevre Dream

Enter number of pages of the book: 350

Enter the category of the book: Horror

Enter a book read by the member: Crime and Punishment

Enter number of pages of the book: 450
```

```
Enter the name of the memeber: Ozu Da
Enter the mobile number of the memeber: 0500000000
Enter the email of the memeber: j@a.a

Enter a book read by the member: The Tatami Galaxy

Enter number of pages of the book: 350

Enter the category of the book: Psychological

Enter a book read by the member: The Beatles

Enter number of pages of the book: 560

Enter the category of the book: Musical

Enter a book read by the member: Hamlet

Enter number of pages of the book: 500

Enter the category of the book: History

Enter a book read by the member: Ok Computer

Enter number of pages of the book: 600

Enter the category of the book: Musical

Enter a book read by the member: 2001

Enter number of pages of the book: 669

Enter the category of the book: Mystery
Do you wish to add another book? (Type Y to add, Type N to exit)
n
Books initilizing done.

Do you wish to add another memeber? (Type Y to add, Type N to exit)
n
Members initilizing done, now exiting.
```

```
Number of books read by the group:
56

Number of Pages read by the group:
28470

<<Categories Ranks By Most Read>>
Category                |         Frequency
-----------------------------------------------------------
Horror                  |            15
Fantasy                 |             8
Mystery                 |             7
Musical                 |             7
Psychological           |             4
Crime                   |             4
Apocalyptic             |             3
Educational             |             3
Sport                   |             2
War                     |             1
Drama                   |             1
History                 |             1
```

```
<<Members Ranks Based on Most Books>>
Memeber                 |       Number Of Books
-----------------------------------------------------------
Nasser Ali              |               7
Sarah Ismail            |               7
Mohammed Fahad          |               6
Joshua Graham           |               6
Thom Yorke              |               5
Isacc Ahmed             |               5
Khalid Bader            |               5
Maryam Mansour          |               5
Doc Mitchel             |               5
Ozu Da                  |               5

<<Members Ranks Based on Most Pages read>>
Member                  |       Number of Pages
-----------------------------------------------------------
Joshua Graham           |             3787
Nasser Ali              |             3755
Sarah Ismail            |             3662
Mohammed Fahad          |             3200
Ozu Da                  |             2679
Doc Mitchel             |             2620
Thom Yorke              |             2350
Isacc Ahmed             |             2325
Maryam Mansour          |             2288
Khalid Bader            |             1804

C:\Users\Pc\source\repos\ConsoleApp1\ConsoleApp1\bin\Debug\netcorea
 0.
To automatically close the console when debugging stops, enable Tool
e when debugging stops.
Press any key to close this window . . .
```