Data structures used in software application with justifications why the data structures are selected

In this project arrays were used as the main data structures in this software application. Arrays are linear data structures consisting of a sequence of items of the same date type stored continuously in computer memory, with its data accessible through the specification of each elements unique index integer number and its major strength being that each element of an array can be accessed in the same amount of time regardless of where in the array the element is located (Levitin, 2012, p. 52).

As the program was relatively simple in its design, being small in size with a probable small total data set and limited functionality, it was deemed appropriate to use arrays. A key reason for the selection of arrays as the main data structures used in the software application was because the primary and fundamental function of the program is to move an object from one array to another, with those objects that are stored in the arrays being instances of the same object types. This compatibility of the different types of arrays and the constant time taken to access each element in the arrays by their index numbers justified their selection as the main data structures for the software application.

In this project arrays were used to store grouped types of “singular” ***Tool*** objects that contained two strings and an integer. Arrays were also used to store a record or register of ***Customer*** objects. These ***Customer*** objects contained a string and an integer plus an aforementioned ***Tool*** object. The design of the program planned for the ***Tool*** objects to be created, deleted and most importantly moved in and out, backwards and forwards from each of these types of arrays. Because the program only dealt with moving the same data types and the object that they were a part of (***Tool***) it was deemed appropriate to use the arrays as the foundational data structures in this project. This would also provide simplicity in the basic operations of the program design. The use of the arrays in the project were also considered suitable as the main data structure in regards to the scope and brief of the assignment task provided.

If this project were to be extended and then to expand to bigger data sets and/or designing a more complex inventory management system, it would be considered possibly appropriate to improve the software program through the use of linked lists as the main data structure. The operation of insertion and deletion of elements over an array are problematic as to insert or delete anywhere in the array, it is necessary to traverse the array and then shift remaining elements as per the operation (Eshitadutta, 2010, para. 11). This is a problem that linked lists do not have, and in a larger project this should be an important consideration. This may not be a major issue in the functionality of this particular program in this project, however if it were to expand, using arrays could limit the time efficiency of the basic operations in these algorithms and hence affect the optimization of a larger and more complex program.

One advantage that using arrays has is that arrays require the same time to access each and every one of its elements which is a positive feature that distinguishes arrays from linked lists (Levitin, 2012, p. 52). This is particularly notable in this project as user input and choice are the guide for searching the data in the program. The data sets are small, and the disadvantages of arrays are minor in comparison to the advantages. Despite the possible use of linked lists in a more advanced and a potential longer version of a similar project it is still concluded that the use of arrays as the major data structure in this software application is justified and appropriate for the functionality of this program.

Algorithm used, and how it is used, for implementing the function 'add new tools to the inventory management system

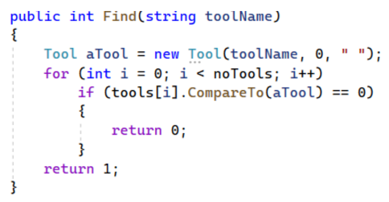
At the beginning of the program 9 separate objects of the ***ToolCatergorie***s class are instantiated. These objects are essentially arrays of ***Tool*** objects. They are designed with the purpose of grouping tools in the inventory management system into 9 different categories that correspond with the 9 types of tools in the tool library. These are; gardening tools, flooring tools, fencing tools, measuring tools, cleaning tools, painting tools, electronic tools, electricity tools and automotive tools. These are also known as tool arrays or tool type arrays in this report. Once the user has selected the option to add a new tool to the inventory management system, they are prompted to choose which of these 9 tool types that they wish to add the tool to. Once they have selected the tool type the corresponding tool array is then used to insert the new ***Tool*** object into. The user is then prompted to enter the name of the tool they wish to add to the tool library.

The ***Find()*** method from the ***ToolCatergories*** class is then used to check if the tool name provided from user input is already in the tool type array. The ***Find()*** method is used to call the ***CompareTo()*** method in the ***Tool*** class. As the method iterates through the array of tools, it compares the tools to the ***Tool*** object that is instantiated within the method that passes the parameter of the tool name. If the tool name is the same as that as one that is in the array it will return an integer value of 0 or 1 indicating the result. This is then followed by an ***If statement***. If the tool is in the array, then the user is prompted that it already exists in the inventory management system and that they need to only alter the recorded quantity of the tool. The program then uses the ***FindIndex()*** method from the ***ToolCatergories*** class. This method uses the same algorithm as the previously called ***Find()*** method, in that it also uses the ***CompareTo()*** method from the ***Tool*** class through and iteration of the array. The difference with this method, however, is when a comparison is made and the tool name is found in the array, instead of returning an integer to indicate the existence of that tool name, instead the index number of that array element is then returned. This then provides the program with the index number in the form of an integer variable which can then be used to find the array element that has that tool name when it is required.

In the next step of the program the index number for the tool is passed into the parameter of the ***Addquantity()*** method. This method allows the user to increase or decrease the tool quantity property of a **Tool** object in an array. This is done by a simple increment or decrement of the array element tool quantity with the corresponding index number that is passed to the method. Once this is completed the ***Tool*** object in the array has been altered and the inventory management system updated.

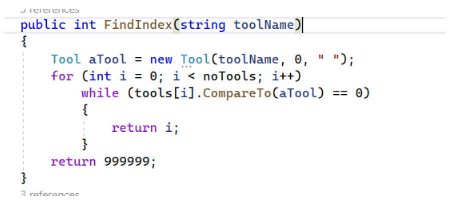
Conversely to this outcome, in the case of the ***Find()*** method indicating that the tool name provided by the user is not found in the array the original ***If statement*** is not met. If the tool is found not to already exist in the inventory management system, the user is then prompted to enter the description and quantity for the tool. This then provides the constructor with the required properties to instantiate the new ***Tool*** object. The tool name, tool description and tool quantity are then passed to the ***Insert()*** method of the ***ToolCatergories*** class as parameters. This method then instantiates the new ***Tool*** object with the parameters then adds them to the array and proceeds to increment the index numbers available in the array. The new tool has then been added to the array of tools of that tool type.

Find() method and algorithm



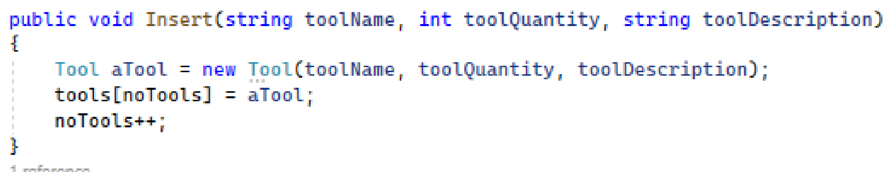
```
public int Find(string toolName)
{
    Tool aTool = new Tool(toolName, 0, " ");
    for (int i = 0; i < noTools; i++)
        if (tools[i].CompareTo(aTool) == 0)
        {
            return 0;
        }
    return 1;
}
```

FindIndex() method and algorithm



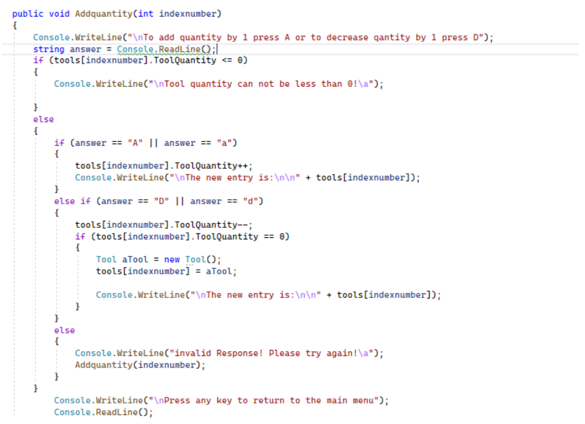
```
public int FindIndex(string toolName)
{
    Tool aTool = new Tool(toolName, 0, " ");
    for (int i = 0; i < noTools; i++)
        while (tools[i].CompareTo(aTool) == 0)
        {
            return i;
        }
    return 999999;
}
```

Insert() method and algorithim



```
public void Insert(string toolName, int toolQuantity, string toolDescription)
{
    Tool aTool = new Tool(toolName, toolQuantity, toolDescription);
    tools[noTools] = aTool;
    noTools++;
}
```

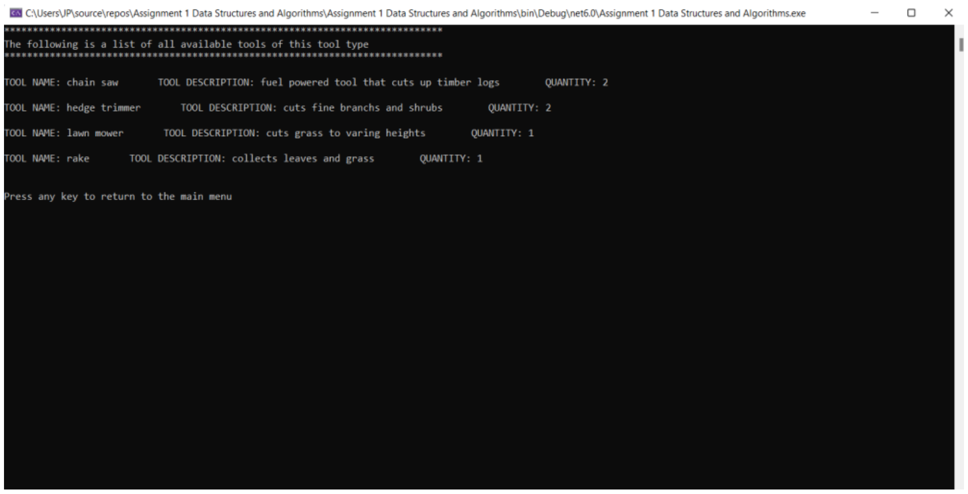
Addquantity() method and algorithm



```
public void Addquantity(int indexnumber)
{
    Console.WriteLine("\nTo add quantity by 1 press A or to decrease qantity by 1 press D");
    string answer = Console.ReadLine();
    if (tools[indexnumber].ToolQuantity <= 0)
    {
        Console.WriteLine("\nTool quantity can not be less than 0!\a");

    }
    else
    {
        if (answer == "A" || answer == "a")
        {
            tools[indexnumber].ToolQuantity++;
            Console.WriteLine("\nThe new entry is:\n\n" + tools[indexnumber]);
        }
        else if (answer == "D" || answer == "d")
        {
            tools[indexnumber].ToolQuantity--;
            if (tools[indexnumber].ToolQuantity == 0)
            {
                Tool aTool = new Tool();
                tools[indexnumber] = aTool;

                Console.WriteLine("\nThe new entry is:\n\n" + tools[indexnumber]);
            }
        }
        else
        {
            Console.WriteLine("invalid Response! Please try again!\a");
            Addquantity(indexnumber);
        }
    }
        Console.WriteLine("\nPress any key to return to the main menu");
        Console.ReadLine();
```

Functional testing results including screenshots for each of the functional tests

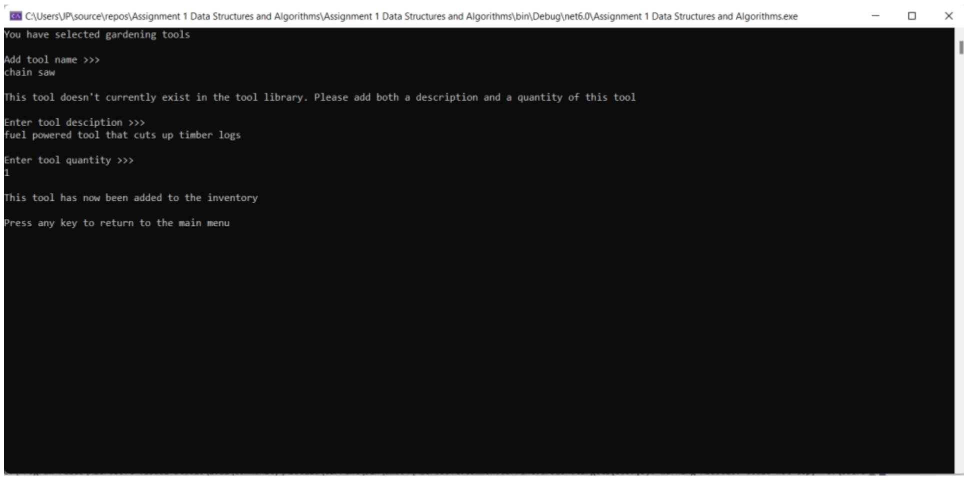
- **Display the information about all the tools of a type selected by the user, including the name of the tool, a brief description of the tool, and the availability of the tool in alphabetical order by the tool names**



```
C:\Users\JP\source\repos\Assignment 1 Data Structures and Algorithms\Assignment 1 Data Structures and Algorithms\bin\Debug\net6.0\Assignment 1 Data Structures and Algorithms.exe
*****************************************************************************
The following is a list of all available tools of this tool type
*****************************************************************************

TOOL NAME: chain saw       TOOL DESCRIPTION: fuel powered tool that cuts up timber logs        QUANTITY: 2

TOOL NAME: hedge trimmer       TOOL DESCRIPTION: cuts fine branchs and shrubs        QUANTITY: 2

TOOL NAME: lawn mower       TOOL DESCRIPTION: cuts grass to varing heights        QUANTITY: 1

TOOL NAME: rake       TOOL DESCRIPTION: collects leaves and grass        QUANTITY: 1


Press any key to return to the main menu
```

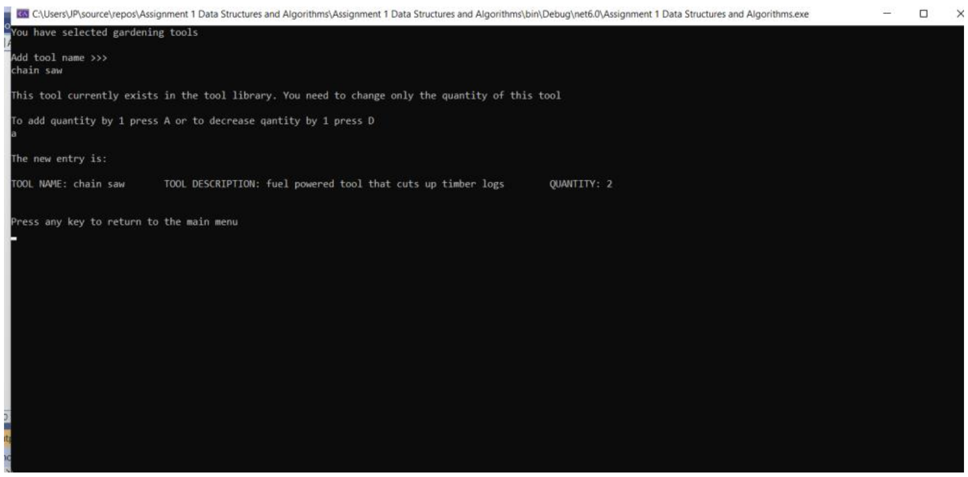
- **Add new tools to the inventory management system**

This screenshot is produced from the user choosing a tool type to add a new tool to and then adding a tool to the library that is not already in the inventory management system. The user is prompted to enter a description of the new tool as well as the quantity of the tool. The tool is then added to the tool library.



```
C:\Users\JP\source\repos\Assignment 1 Data Structures and Algorithms\Assignment 1 Data Structures and Algorithms\bin\Debug\net6.0\Assignment 1 Data Structures and Algorithms.exe
You have selected gardening tools

Add tool name >>>
chain saw

This tool doesn't currently exist in the tool library. Please add both a description and a quantity of this tool

Enter tool desciption >>>
fuel powered tool that cuts up timber logs

Enter tool quantity >>>
1

This tool has now been added to the inventory

Press any key to return to the main menu
```

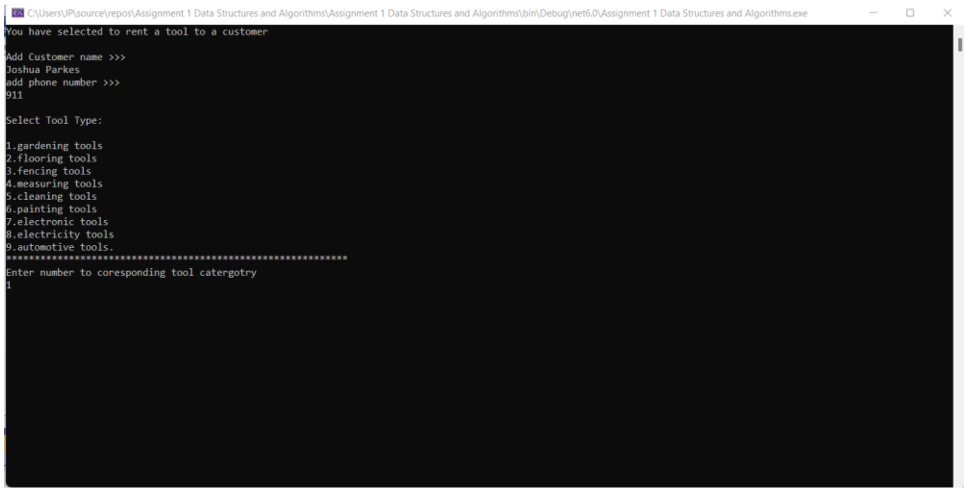
If the tool name entered by the user is already in inventory, only quantity of the tool is changed and recorded in the inventory management system.



```
C:\Users\JP\source\repos\Assignment 1 Data Structures and Algorithms\Assignment 1 Data Structures and Algorithms\bin\Debug\net6.0\Assignment 1 Data Structures and Algorithms.exe
You have selected gardening tools

Add tool name >>>
chain saw

This tool currently exists in the tool library. You need to change only the quantity of this tool

To add quantity by 1 press A or to decrease qantity by 1 press D
a

The new entry is:

TOOL NAME: chain saw       TOOL DESCRIPTION: fuel powered tool that cuts up timber logs        QUANTITY: 2


Press any key to return to the main menu
```

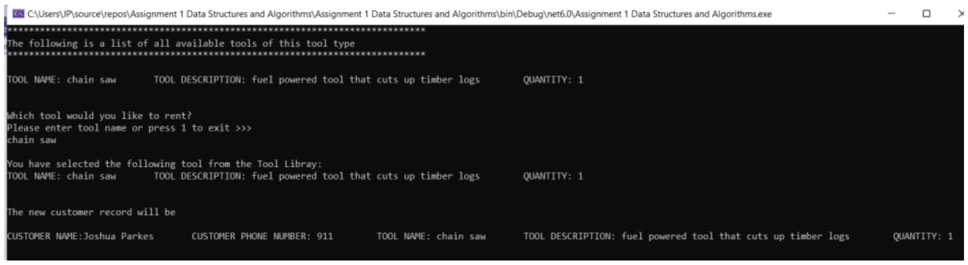
- **Lend a tool in the tool library to a person**

The screenshot below is produced from option 2 on the main menu, renting a tool out to a customer. The user is prompted for the customer's name and phone number. The user is then prompted for the tool type that the customer requires.



```
C:\Users\JP\source\repos\Assignment 1 Data Structures and Algorithms\Assignment 1 Data Structures and Algorithms\bin\Debug\net6.0\Assignment 1 Data Structures and Algorithms.exe
You have selected to rent a tool to a customer

Add Customer name >>>
Joshua Parkes
add phone number >>>
911

Select Tool Type:

1.gardening tools
2.flooring tools
3.fencing tools
4.measuring tools
5.cleaning tools
6.painting tools
7.electronic tools
8.electricity tools
9.automotive tools.
************************************************************
Enter number to coresponding tool catergotry
1
```

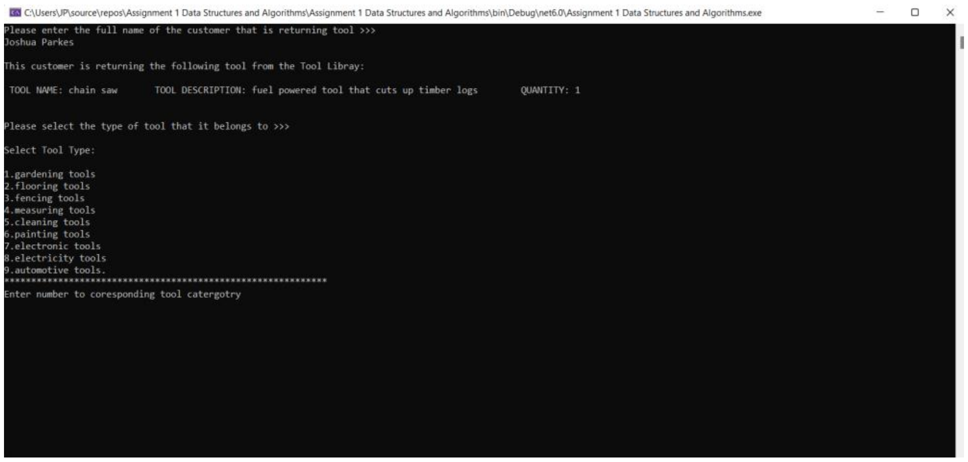
A list of the tools in the selected tool type is then displayed for the user and they are prompted to enter the name of the tool they wish to rent out. The screen then displays the tool selected by the user and the new customer record.



```
C:\Users\JP\source\repos\Assignment 1 Data Structures and Algorithms\Assignment 1 Data Structures and Algorithms\bin\Debug\net6.0\Assignment 1 Data Structures and Algorithms.exe
*****************************************************************************
The following is a list of all available tools of this tool type
*****************************************************************************

TOOL NAME: chain saw       TOOL DESCRIPTION: fuel powered tool that cuts up timber logs       QUANTITY: 1


Which tool would you like to rent?
Please enter tool name or press 1 to exit >>>
chain saw

You have selected the following tool from the Tool Libray:
TOOL NAME: chain saw       TOOL DESCRIPTION: fuel powered tool that cuts up timber logs       QUANTITY: 1


The new customer record will be

CUSTOMER NAME:Joshua Parkes       CUSTOMER PHONE NUMBER: 911       TOOL NAME: chain saw       TOOL DESCRIPTION: fuel powered tool that cuts up timber logs       QUANTITY: 1
```

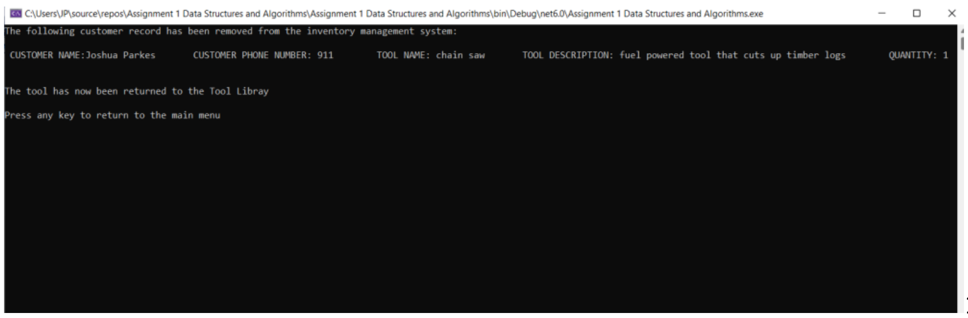
- **Return a rented tool to the tool library**

In the screenshot below the user selects to return a tool to the tool library from the main menu and is prompted to enter the full name of the customer that is returning the tool. The tool the customer is returning is displayed and the user is prompted to select the tool type that the tool belongs to.



```
C:\Users\JP\source\repos\Assignment 1 Data Structures and Algorithms\Assignment 1 Data Structures and Algorithms\bin\Debug\net6.0\Assignment 1 Data Structures and Algorithms.exe
Please enter the full name of the customer that is returning tool >>>
Joshua Parkes

This customer is returning the following tool from the Tool Libray:

 TOOL NAME: chain saw       TOOL DESCRIPTION: fuel powered tool that cuts up timber logs       QUANTITY: 1


Please select the type of tool that it belongs to >>>

Select Tool Type:

1.gardening tools
2.flooring tools
3.fencing tools
4.measuring tools
5.cleaning tools
6.painting tools
7.electronic tools
8.electricity tools
9.automotive tools.
************************************************************
Enter number to coresponding tool catergotry
```

Confirmation of the old customer record is displayed before it is removed from the inventory management system. Confirmation is then given about the tool being returned to the tool library.

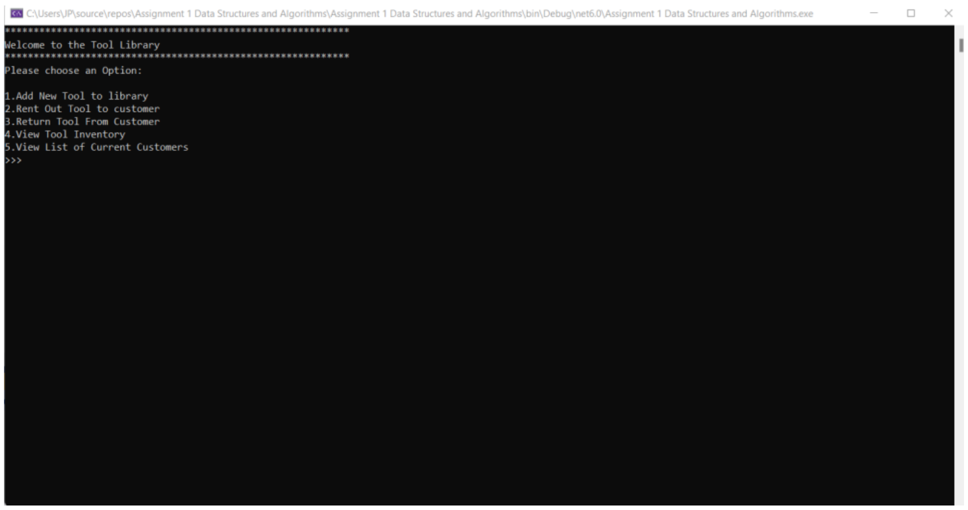


```
C:\Users\JP\source\repos\Assignment 1 Data Structures and Algorithms\Assignment 1 Data Structures and Algorithms\bin\Debug\net6.0\Assignment 1 Data Structures and Algorithms.exe
The following customer record has been removed from the inventory management system:

 CUSTOMER NAME:Joshua Parkes       CUSTOMER PHONE NUMBER: 911       TOOL NAME: chain saw       TOOL DESCRIPTION: fuel powered tool that cuts up timber logs       QUANTITY: 1


The tool has now been returned to the Tool Libray

Press any key to return to the main menu
```

1

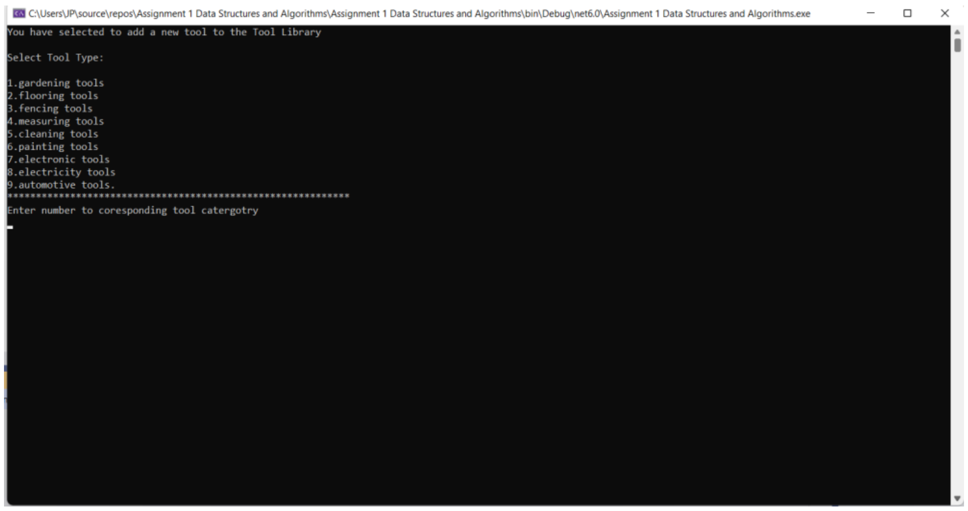
- **A command-line user interface with a functionality menu.**

The main menu contains 5 options for the user.


C:\Users\JP\source\repos\Assignment 1 Data Structures and Algorithms\Assignment 1 Data Structures and Algorithms\bin\Debug\net6.0\Assignment 1 Data Structures and Algorithms.exe

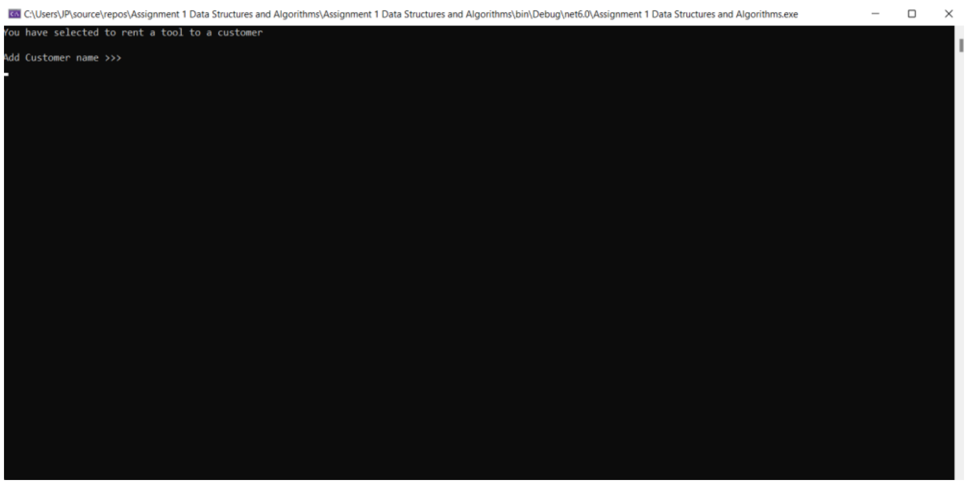
```
************************************************************
Welcome to the Tool Library
************************************************************
Please choose an Option:

1.Add New Tool to library
2.Rent Out Tool to customer
3.Return Tool From Customer
4.View Tool Inventory
5.View List of Current Customers
>>>
```

Option 1: Add New Tool to Library


C:\Users\JP\source\repos\Assignment 1 Data Structures and Algorithms\Assignment 1 Data Structures and Algorithms\bin\Debug\net6.0\Assignment 1 Data Structures and Algorithms.exe

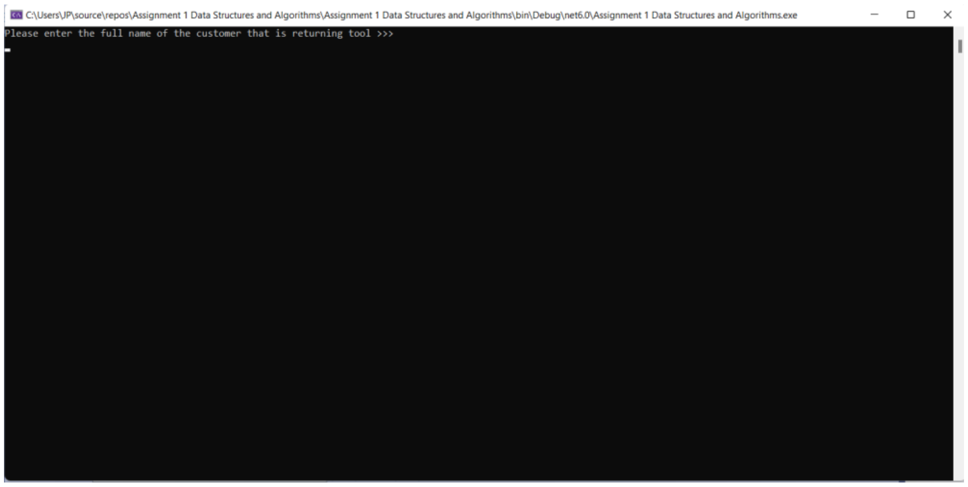
```
You have selected to add a new tool to the Tool Library

Select Tool Type:

1.gardening tools
2.flooring tools
3.fencing tools
4.measuring tools
5.cleaning tools
6.painting tools
7.electronic tools
8.electricity tools
9.automotive tools.
************************************************************
Enter number to coresponding tool catergotry
```

Option 2: Rent Out Tool to Customer


C:\Users\JP\source\repos\Assignment 1 Data Structures and Algorithms\Assignment 1 Data Structures and Algorithms\bin\Debug\net6.0\Assignment 1 Data Structures and Algorithms.exe

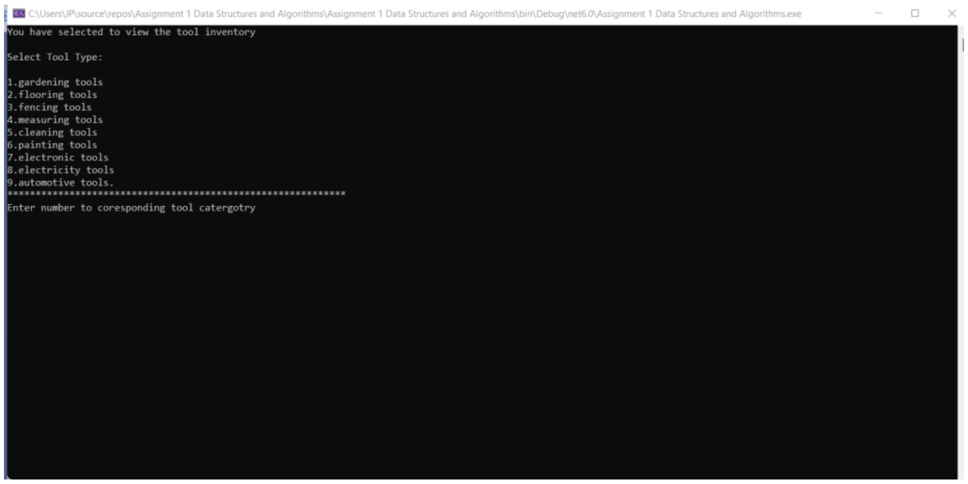
```
You have selected to rent a tool to a customer

Add Customer name >>>
```

Option 3: Return Tool from Customer



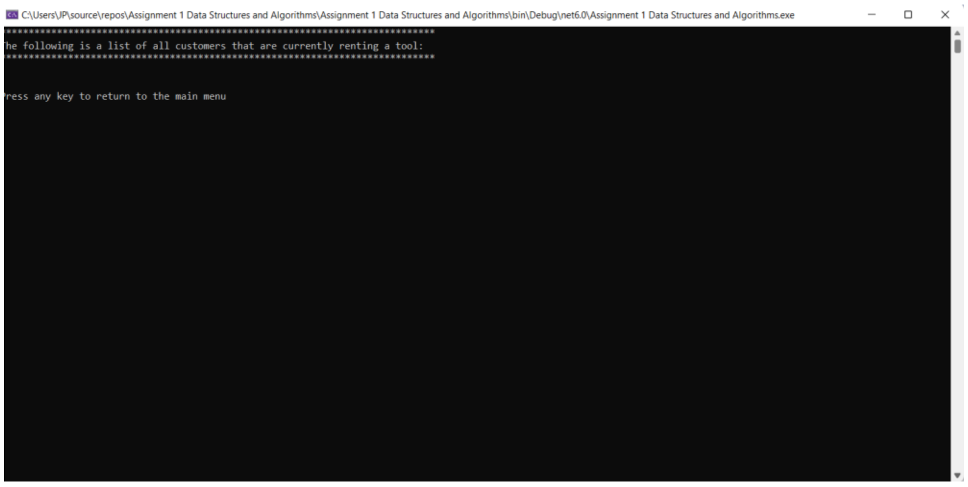
```
C:\Users\JP\source\repos\Assignment 1 Data Structures and Algorithms\Assignment 1 Data Structures and Algorithms\bin\Debug\net6.0\Assignment 1 Data Structures and Algorithms.exe
Please enter the full name of the customer that is returning tool >>>
```

Option 4: View Tool Inventory



```
C:\Users\JP\source\repos\Assignment 1 Data Structures and Algorithms\Assignment 1 Data Structures and Algorithms\bin\Debug\net6.0\Assignment 1 Data Structures and Algorithms.exe
You have selected to view the tool inventory

Select Tool Type:

1.gardening tools
2.flooring tools
3.fencing tools
4.measuring tools
5.cleaning tools
6.painting tools
7.electronic tools
8.electricity tools
9.automotive tools.
************************************************************
Enter number to coresponding tool catergotry
```

Option 5: View List of Current Customers



```
C:\Users\JP\source\repos\Assignment 1 Data Structures and Algorithms\Assignment 1 Data Structures and Algorithms\bin\Debug\net6.0\Assignment 1 Data Structures and Algorithms.exe
****************************************************************************
he following is a list of all customers that are currently renting a tool:
****************************************************************************


ress any key to return to the main menu
```

References

Levitin, A. (2012). *Introduction to the design and analysis of algorithms* (3rd ed.). Pearson.

Eshitadutta. (2022). *Advantages and Disadvantages of Array in C*. GeeksforGeeks. https://www.geeksforgeeks.org/advantages-and-disadvantages-of-array-in-c/