**Arithmetic Challenge Technical Documentation 2018**

Contents

[User Manual 2](#_Toc524950026)

[UML Diagrams 5](#_Toc524950027)

[Sorting Algorithms 8](#_Toc524950028)

[Source Control 9](#_Toc524950029)

[Emails 10](#_Toc524950030)

[Debugging 12](#_Toc524950031)

## User Manual

This application was developed to allow teachers to send equations out to a single student and wait for their answer.

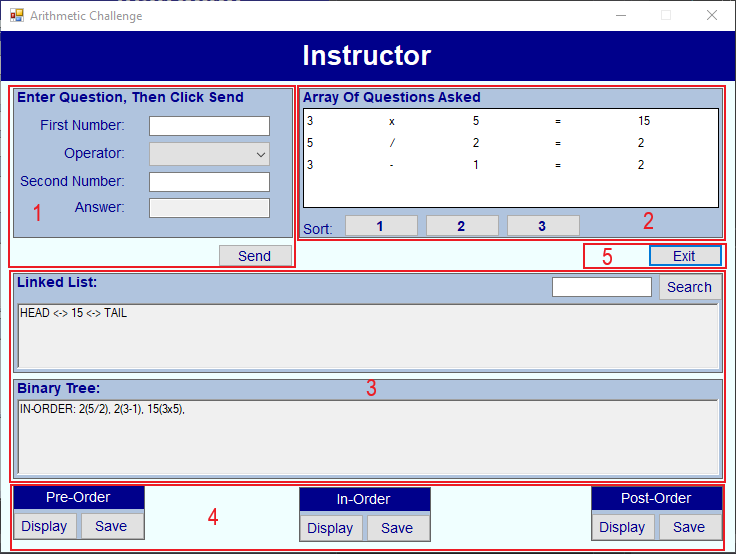


Image 1 – Arithmetic Challenge Server

**Image 1 – Section 1:** This section contains the question details to send to the student.

1. First number
2. Operators (+, -, x, /)
3. Second number
4. Answer
5. Send – Sends the question details (Except for the answer of course) to the student.

**Image 1 – Section 2:** This section contains the array of questions asked and sorting options.

1. The array of questions asked displayed in a grid format.
2. Sort 1 – Sorts the list of questions from smallest to largest by answer.
3. Sort 2 – Sorts the list of questions from largest to smallest by answer.
4. Sort 3 – Sorts the list of questions by operators.

**Image 1 – Section 3:** This section contains the linked list of all incorrectly answered questions, the binary tree of all questions asked and a search option for the linked list.

1. The rich text box adds the answers that are answered incorrectly by the student.
2. The rich text box for the binary tree adds all the questions sent to the student to the binary tree.
3. Search – When an answer is entered in the textbox it does a binary search of all the answers in the linked list and returns found or not found.

**Image 1 – Section 4:** This section contains different ordering options of the binary tree and save options.

Pre-Order:

1. Display – Displays the binary tree in the rich text box pre-ordered.
2. Save – Saves the Pre-Ordered binary tree to a text file.

In-Order:

1. Display – Displays the binary tree in the rich text box in-order.
2. Save – Saves the In-Order binary tree to a text file.

Post-Order:

1. Display – Displays the binary tree in the rich text box post-order.
2. Save – Saves the Post-Ordered binary tree to a text file.

**Image 1 – Section 5:** This section contains an option to close the application.

1. Exit – Exits the application.

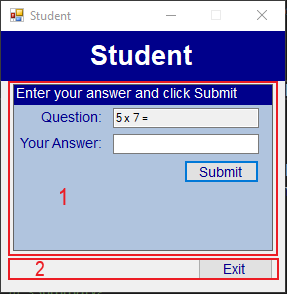
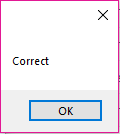
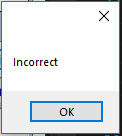


Image 2 – Arithmetic Challenge Student

**Image 2 – Section 1:** This section contains the question asked from the server (teacher), an answer input and a submit button.

1. Question – The question asked from the server displays in the textbox.
2. Your Answer – The student enters an answer to the question in the textbox.
3. Submit – This button sends the students answer to the question back to the server (You must answer all questions!).

If the answer is correct a message will display to let the student know. 

If the answer is incorrect a message will display to let the student know. 

**Image 2 – Section 2:** This section contains an option for the student to close the application.

1. Exit – Exits the application.

## UML Diagrams

**Arithmetic Challenge Server:**

|  |
| --- |
| **Form 1** |
| SetTextCallback(string text): delegate void  Listener: TcpListener  Client: TcpClient  ns: NetworkStream  t: Thread  listOfNodes: NodeList  myTree: BinaryTree  equation: Question |
| buttonExit\_Click(object sender, EventArgs e)  textChanges(object sender, EventArgs e)  autoAnswer()  saveEquation()  buttonSend\_Click(object sender, EventArgs e)  DoWork()  SetText(string text)  binaryTree()  clearEquation()  dataGridQuestions\_SelectionChanged(object sender, EventArgs e)  buttonSearch\_Click(object sender, EventArgs e)  button1\_Click(object sender, EventArgs e)  buttonSort2(object sender, EventArgs e)  buttonPreDisplay(object sender, EventArgs e)  buttonInDisplay(object sender, EventArgs e)  buttonPostDisplay(object sender, EventArgs e)  buttonSort3(object sender, EventArgs e)  buttonPreSave\_Click(object sender, EventArgs e)  buttonInSave\_Click(object sender, EventArgs e)  buttonPostSave\_Click(object sender, EventArgs e) |

|  |
| --- |
| **Question** |
| firstNumber: public int  secondNumber: public int  Symbol: public string  answer: public int |
| Question(int firstNum, int secondNum, string symbol, int result) |

|  |
| --- |
| **Node** |
| myValue: int  previous, next: Node |
| Node(int aValue)  getValue()  setMyValue(int aValue)  setNext(Node aNode)  getNext()  setPrevious(Node aNode)  getPrevious()  tostring() |

|  |
| --- |
| **NodeList** |
| CurrentNode, HeadNode, TailNode: Node  count: static int |
| NodeList()  NodeList(Node aNode)  getCurrentNode()  getHeadNode()  getTailNode()  setCurrentNode(Node aNode)  setHeadNode(Node aNode)  setTailNode(Node aNode)  addAtFrontOfNodeList(Node aNode)  SortList()  binarySearch(int searchValue)  printLinkList(NodeList aNode)  linkListTable(NodeList aNode) |

|  |
| --- |
| **BinaryNode** |
| Equation: Question  left: BinaryNode  right: BinaryNode |
| BinaryNode(Question val)  NodeToString() |

|  |
| --- |
| **BinaryTree** |
| top: BinaryNode  printStr: private static string |
| printPreOrder(BinaryTree tree)  PreOrder(BinaryNode Root)  printInOrder(BinaryTree tree)  InOrder(BinaryNode Root)  printPostOrder(BinaryTree tree)  PostOrder(BinaryNode Root)  BinaryTree(Question answerVal)  BinaryTree()  Add(Question quest) |

**Arithmetic Challenge Client:**

|  |
| --- |
| **Form 1** |
| portNum: private cons tint  SetTextCallback(string text): delegate void  client: TcpClient  ns: NetworkStream  t: Thread  answer: int  hostName: private const string |
| buttonExit\_Click(object sender, EventArgs e)  buttonSubmit\_Click(object sender, EventArgs e)  DoWork()  SetText(string text) |

## Sorting Algorithms

**Bubble Sort:**

Advantages – It is straight forward and a simplistic method to being able to sort data.

Disadvantage – Other sorting methods have better performance than bubble sort such as the insertion sort.

**Quick Sort (Partition Sort):**

Advantages – Strong performance and easily implemented. Its cache performance is higher than other sorting algorithms.

Disadvantages – It is unstable and heavily decreases in speed down to O(n2) in the case of unsuccessful pivot selections.

**Insertion Sort:**

Advantages – It is efficient for small datasets.

Disadvantages – It is not as efficient on large datasets compared to advanced sorting algorithms such as quick sort.

## Source Control

I used SourceTree to manage my changes to the project. This application manages the projects files which is stored in a repository on GitHub.

I committed changes to the master branch of the repository using SourceTree with appropriate descriptions of the changes committed.

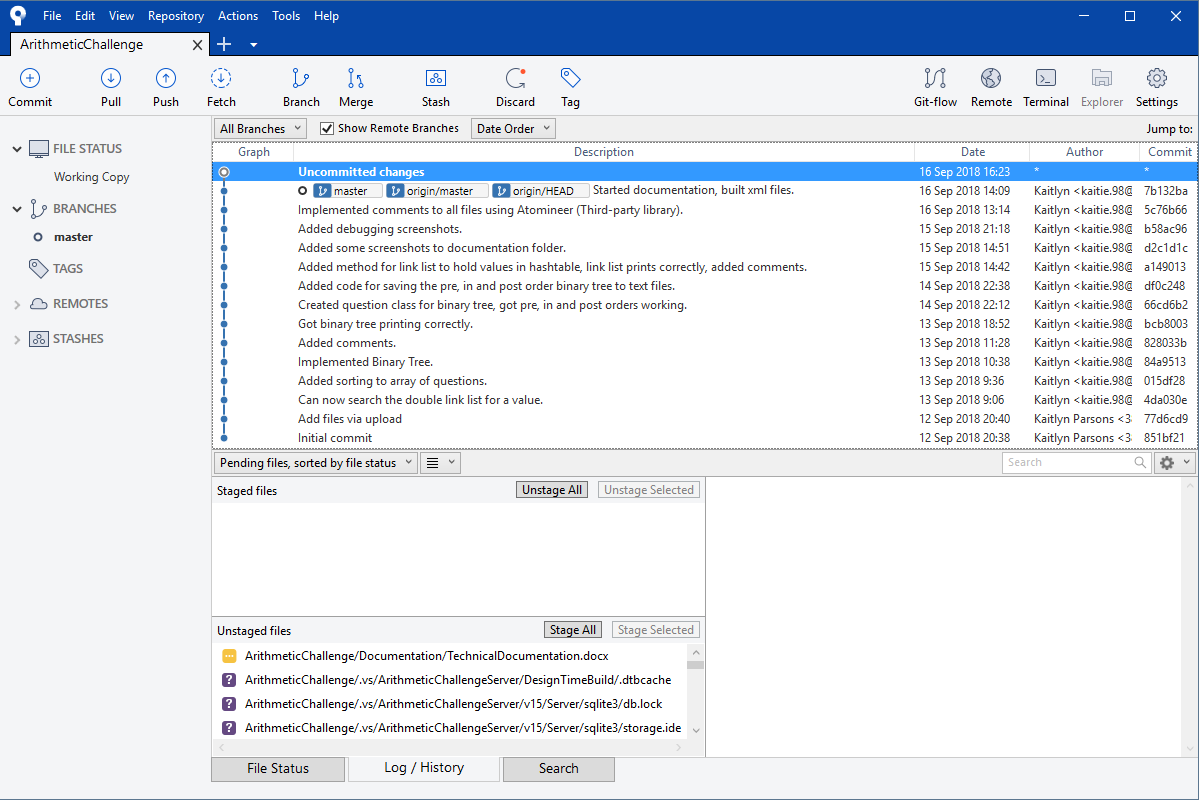


Image 3 – Source Control

## Emails

**Manager Email:**

Hi David,

We will be using threading to send the questions from the server to the student. Threading is the fasted option in sending data between devices.

We are also using a third party library called Atomineer (documenting tool) to save time producing effective comments in the code. This will allow for easier navigation for the developer if the client wants us to implement any new features to the application in the future.

Regards,

Kaitlyn Parsons.

**Client Email:**

Hello Client,

We will be using threading for this application which is a fast and efficient way of sending data between devices.

We are also using a third party library called Atomineer which is a documenting tool. The reason we are using this is to save time producing effective comments within the code behind the application. This will save time and allow for faster implementation of new features.

Regards,

Kaitlyn Parsons.

**Closing Manager Email:**

Hi David,

We have successfully deployed the Arithmetic Challenge application into the production environment and it is ready to use.

We are concerned that the application only allows for connecting a single student to the server. If the client wishes to allow for multiple students to connect to the application we would have to look at implementing that into the application.

Otherwise, the application has been very successful!

Any concerns or queries please don’t hesitate to contact me.

Regards,

Kaitlyn Parsons.

**Closing Client Email:**

Hello Client,

We have successfully launched your application to the production environment.

You can now download and install your application on your device. Please read the user manual to assist you with using the application.

Congratulations on the application!

Any concerns or queries please don’t hesitate to contact me.

Regards,

Kaitlyn Parsons.

## Debugging

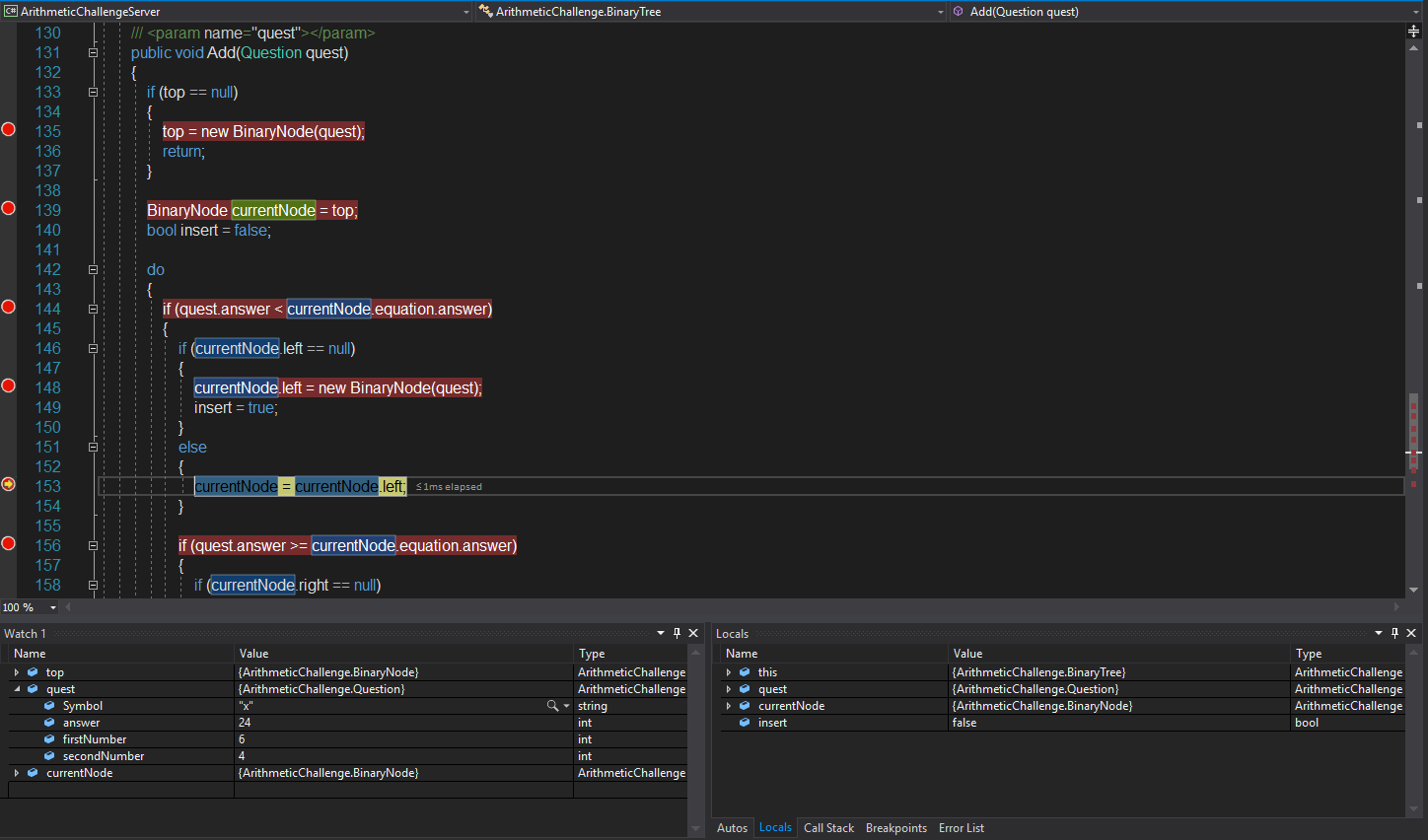


Image 4 – Debugging on the Binary Tree

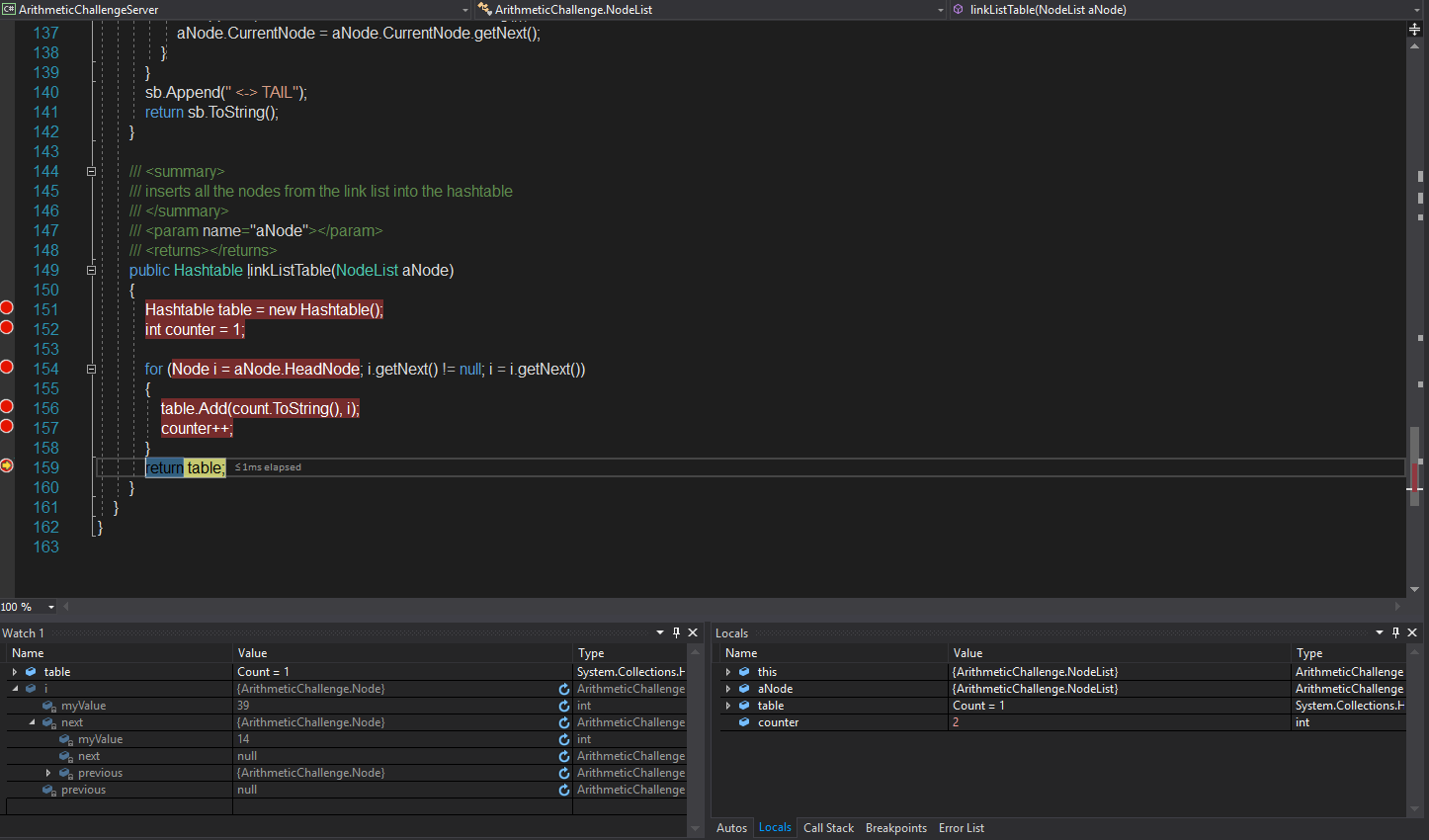


Image 5 – Debugging on the Hash Table method for the Link List