PSP1.1 Project Plan Summary

Name:	James Small		
Program:	3B	Number:	5
Instructor:	Dr. Concepcion	Language:	C++

Summary	**Plan**	**Actual**	**To Date**
LOC/Hour	106.7	66.8	0
Planned Time	45		375
Actual Time		115	498
CPI (Cost-Performance Index)			0.753 (Planned/Actual)
% Reused	0	0	0
% New Reusable	33.8	39.1	0

Program Size (LOC)	**Plan**	**Actual**	**To Date**
Base (B)	151 (Measured)	151 (Measured)	
Deleted (D)	0 (Estimated)	0 (Counted)	
Modified (M)	5 (Estimated)	13 (Counted)	
Added (A)	75 (N-M)	115 (T-B+D-R)	
Reused (R)	0 (Estimated)	0 (Counted)	0
New and Changed (N)	80 (Estimated)	128 (A+M)	0
Total LOC (T)	226 (N+B-M-D+R)	266 (Measured)	749
Total New Reusable	27	50	50

Time in Phase (min.)	**Plan**	**Actual**	**To Date**	**To Date %**
Planning	2	3	18	3.6
Design	4	10	48	9.6
Code	16	49	183	36.7
Compile	4	5	35	7
Test	15	32	157	31.5
Postmortem	5	16	57	11.4
Total	45	115	498	100

Defects Injected	**Actual**	**To Date**	**To Date %**
Planning	0	0	0
Design	0	1	5.9
Code	3	16	94.1
Compile	0	0	0
Test	0	0	0
Total Development	3	17	100

Defects Removed	**Actual**	**To Date**	**To Date %**
Planning	0	0	0
Design	0	0	0
Code	0	0	0
Compile	2	8	47.1
Test	1	9	52.9
Total Development	3	17	100
After Development			

Test Report Template

Name: James Small

Program: 3B Number: 5

Instructor: Dr. Concepcion Language: C++

Test Name/Number	1
Test Objective	To determine if invalid entries on the main menu screen will cause the program to crash or to produce unexpected results.
Test Description	1. Enter a letter when a number is required 2. Enter a number outside the accepted range of 0 to 3 3. Enter multiple characters 4. Enter multiple characters starting with an accepted digit
Test Conditions	
Expected Results	The expected results will say "Invalid Choice, Try Again" for test conditions 1,2, and 3 above. When test case 4 is ran, it will read the first valid accepted digit, in this case a 1, and proceed to the appropriate menu choice for a 1.
Actual Results	The actual results matched the expected results. "Invalid Choice, Try Again" was displayed for test conditions 1 to 3 and a valid choice was accepted for test condition 4. See output example for test 1.

Test Name/Number	2
Test Objective	To determine if in write mode, will invalid entries cause the program to crash or have unexpected results.
Test Description	1. Enter write mode 2. Name File 3. Enter a letter for amount of numbers to write. 4. Enter a negative number for amount of numbers to write. 5. Enter a valid number 6. Enter a letter for the first number 7. Enter a string of letters for the first number. 8. Enter a valid number
Test Conditions	
Expected Results	The expected results will say "Invalid Number, Try Again" when entering the amount of numbers to write on items 3 and 4 above. Item 5 will succeed and it will now ask for the 1st number. Items 6 and 7 will fail above with the following message, "Invalid Value, Try Again". It will succeed when item 8 above is ran.
Actual Results	The actual results matched the expected results. "Invalid Number, Try Again" was displayed when it was supposed to. "Invalid Value, Try Again" was displayed when it was supposed to. The entry succeeded when it was valid.

Test Name/Number	3
Test Objective	To determine if invalid entries on the modify menu screen will cause the program to crash or produce unexpected results.
Test Description	1. Enter modify mode 2. Enter file name to modify 3. Enter a letter for the menu choice 4. Enter a string of letters for the menu choice 5. Enter a number outside of the accepted range of 1 to 5 6. Enter a valid number and will proceed to next number
Test Conditions	
Expected Results	The expected results will say "Invalid Choice, Try Again" for items 3, 4, and 5 above. Items 1, 2, and 6 will work correctly and proceed.
Actual Results	The actual results matched the expected results. "Invalid Choice, Try Again" was displayed when it was supposed to and prevented invalid entries.

Process Improvement Proposal

Name: James Small

Program: 3B

Number: 5

Instructor: Dr. Concepcion

Language: C++

Problem Description Briefly describe the problems that you encountered.
I noticed after finishing up that some of the code I was writing seemed repetitive.
Proposal Description Briefly describe the process improvements that you propose.
A way I could have solved this was to think about it more in the design phase and come up with a few more resumable classes to prevent repetitive code typing and to also add more to my reuse library.
Other Notes and Comments Note any other comments or observations that describe your experiences or improvement ideas.

Size Estimating Template

Name:	James Small		
Program:	3B	Number:	5
Instructor:	Dr. Concepcion	Language:	C++

BASE PROGRAM LOC	ESTIMATE	ACTUAL
BASE SIZE (B)	151	151
LOC DELETED (D)	0	0
LOC MODIFIED (M)	5	13

OBJECT LOC

BASE ADDITIONS	TYPE	METHODS	REL. SIZE	LOC	LOC
TOTAL BASE ADDITIONS (BA)				0	0

NEW OBJECTS	TYPE	METHODS	REL. SIZE	LOC (New Reuse*)	
StringToFloat	Data	3	Medium	27	50*
TOTAL NEW OBJECTS				27	50

REUSED OBJECTS

REUSED TOTAL	0	0

		SIZE	TIME
PROBE Estimating Method:		C	C
Estimated Object LOC (E):	E=BA+NO+M	32	
Regression Parameters:	β_0(size and time)	0	0
Regression Parameters:	β_1(size and time)	1.26047	1.15663
Estimated New and Changed LOC (N):	$N=\beta_0+\beta_1*E$	40.3	
Estimated Total LOC:	T=N+B-D-M+R	186.3	
Estimated Total New Reuse (sum of * LOC):		27	
Estimated Total Development Time:	$Time=\beta_0+\beta_1*E$		37
Prediction Range:	Range	20	20
Upper Prediction Interval:		60.3	57
Lower Prediction Interval:		20.3	17
Prediction Interval Percent:		N/A	N/A

Task Planning Template

Name: James Small

Program: 3B Number: 5

Instructor: Dr. Concepcion Language: C++

Task		Plan					Actual		
#	Name	Minutes	Planned Value	Cumulative Minutes	Cumulative Planned Value	Date	Date	Earned Value	Cumulative Earned Value
1	Planning	2	3.9	2	3.9	2014-02-14	2014-02-14	3.9	3.9
2	Design	4	9.9	6	13.8	2014-02-14	2014-02-14	9.9	13.8
3	Code	16	35	22	48.8	2014-02-15	2014-02-15	35	48.8
4	Compile	4	7.8	25	56.7	2014-02-15	2014-02-15	7.8	56.7
5	Test	15	32.6	40	89.3	2014-02-15	2014-02-15	32.6	89.3
6	Postmortem	5	10.7	45	100	2014-02-15	2014-02-15	10.7	100
Totals		45	100						

Schedule Planning Template

Name: James Small

Program: 3B Number: 5

Instructor: Dr. Concepcion Language: C++

		Plan			Actual		
No.	**Date**	**Direct Minutes**	**Cumulative Minutes**	**Cumulative Planned Value**	**Direct Minutes**	**Cumulative Minutes**	**Cumulative Earned Value**
1	2014-02-14	6	6	13.8	13	13	13.8
2	2014-02-15	39	45	100	102	115	100

Time Recording Log

Name: James Small

Program: 3B Number: 5

Instructor: Dr. Concepcion Language: C++

Date	Start	Stop	Int. Time	Delta Time	Phase	Comments
2014-02-14	12:22	12:25	0	3	Planning	Made time and loc guesses
2014-02-14	12:25	12:35	0	10	Design	Made design docs with new added class
2014-02-15	19:58	20:47	0	49	Code	Wrote code in C++
2014-02-15	20:47	20:52	0	5	Compile	Fixed a few small errors
2014-02-15	20:52	21:24	0	32	Test	Tested and fixed any errors found
2014-02-15	21:24	21:40	0	16	Postmortem	Filled out paperwork

Total: 115

Defect Recording Log

Name: James Small

Program: 3B

Instructor: Dr. Concepcion

Number: 5

Language: C++

Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.
2014-02-15	1	20 - Syntax	Code	Compile	1	

Description: Forgot semicolon at end of do while loop

Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.
2014-02-15	2	20 - Syntax	Code	Compile	5	

Description: when converting char to int with atoi, forgot to pass as reference in multiple locations.

Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.
2014-02-15	3	80 - Function	Code	Test	10	

Description: If user typed multiple characters when single character was required, need to add cin.ignore to stop reading rest of line.

Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.

Description:

Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.

Description:

Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.

Description:

Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.

Description:

```cpp
// Name: James Small
// Program: 3B
// Class: CSE455
// Description: Program to input, output, or modify a file.

#include <iostream>
#include <string>
#include <stdlib.h> // for atoi
#include <ctype.h> // for isdigit
#include "Input.h"

using namespace std;

int main()
{
    char choice = 0;
    bool choiceGood = false;

    do {
        cout << "What would you like to do?\n";
        cout << "Enter 1 to read from file.\n";
        cout << "Enter 2 to write to file.\n";
        cout << "Enter 3 to modify a file.\n";
        cout << "Enter 0 to quit.\n";
        cout << "Choice: ";

        cin >> choice;

        if (isdigit(choice)) {
            if (atoi(&choice) >= 0 && atoi(&choice) < 4)
                choiceGood = true;
            else
                cout << "\nInvalid Choice, Try again\n\n";
        } else
            cout << "\nInvalid Choice, Try again\n\n";

        cin.ignore(INT_MAX,'\n');

    } while (!choiceGood);

    if (choice != '0') {

        string file;

        cout << "Enter the file name to access: ";
        cin >> file;

        Input input(file);

        if (choice == '1')
            input.readFromFile();
        else if (choice == '2')
            input.writeToFile();
        else if (choice == '3')
            input.modifyFile();
    }
    return 0;
}
```

```
// Name: James Small
// Program: 3B
// Class: CSE455
// Description: Input class Header File

#ifndef INPUT_H
#define INPUT_H

#include <string>
#include "StringToFloat.h"

using namespace std;

class Input
{
    public:
        Input(string fileName);
        void writeToFile();
        void readFromFile();
        void modifyFile();

    private :
        string fileToRead;
        float enterNumber();
        StringToFloat stringToFloat;
};
#endif
```

```
// Name: James Small
// Program: 3B
// Class: CSE455
// Description: Input class Implementation File

#include "Input.h"
#include <fstream>
#include <iostream>
#include <vector>
#include <stdlib.h> // for atoi
#include <ctype.h> // for isdigit

using namespace std;

// This is the default constructor

Input::Input(string fileName)
{
    this->fileToRead = fileName;
}

// This method asks user for a set of numbers and outputs them to a file

void Input::writeToFile()
{
    char count = 0;
    float currentValue;
    string currentString = "";
    bool countGood = false;

    do {
        cout << "Enter the amount of numbers to write: ";

        cin >> count;

        if (isdigit(count)) {
            if (atoi(&count) > 0)
                countGood = true;
            else
                cout << "\nInvalid number, Try again\n\n";
        } else
            cout << "\nInvalid number, Try again\n\n";

        cin.ignore(INT_MAX,'\n');

    } while (!countGood);

    ofstream outfile;

    outfile.open(fileToRead.c_str());

    for (int i = 0; i < atoi(&count); i++) {

        cout << "Enter number " << i + 1 << ": ";

        cin >> currentString;
        while (!stringToFloat.isStringAFloat(currentString)) {

            cout << "\nInvalid Value, try again\n\n";
            cout << "Enter number " << i + 1 << ": ";

            cin.ignore(INT_MAX,'\n');

            cin >> currentString;
        }

        currentValue = stringToFloat.getFloatValue();

        if (i == count - 1)
            outfile << currentValue;
        else
            outfile << currentValue << " ";
    }

    outfile.close();
}

// This method reads in a set of numbers from a file and displays them on screen

void Input::readFromFile()
{
    ifstream infile;

    infile.open(fileToRead.c_str());

    float currentValue = 0;

    while (!infile.eof()) {
        infile >> currentValue;
        cout << currentValue << endl;
    }

    infile.close();
}

// This method modifies an existing file one line at a time.

void Input::modifyFile()
{
    ifstream infile;

    infile.open(fileToRead.c_str());

    float currentValue = 0;
    char choice;
    vector<float> currentNumbers;
    bool acceptAllNumbers = false;

    while (!infile.eof()) {
        infile >> currentValue;
        if (acceptAllNumbers) {
            currentNumbers.push_back(currentValue);
        } else {

            bool choiceGood = false;
            do {
                cout << "\nWhat would you like to do with this number, " <<
                    currentValue << "?\n";
                cout << "Enter 1 to accept this number.\n";
                cout << "Enter 2 to replace this number.\n";
                cout << "Enter 3 to delete this number.\n";
                cout << "Enter 4 to insert a new number after current number.\n";
                cout << "Enter 5 to accept the remainder of the numbers.\n";
                cout << "Choice: ";

                cin >> choice;

                if (isdigit(choice)) {
                    if (atoi(&choice) > 0 && atoi(&choice) < 6)
                        choiceGood = true;
                    else
                        cout << "\nInvalid Choice, Try again\n\n";
                } else
                    cout << "\nInvalid Choice, Try again\n\n";

                cin.ignore(INT_MAX,'\n');

            } while (!choiceGood);

            switch (choice) {
                case '1':
                    currentNumbers.push_back(currentValue);
                    break;
                case '2':
                    currentNumbers.push_back(enterNumber());
                    break;
                case '3':
                    break;
                case '4':
                    currentNumbers.push_back(currentValue);
                    currentNumbers.push_back(enterNumber());
                    break;
                case '5':
                    currentNumbers.push_back(currentValue);
                    acceptAllNumbers = true;
                    break;
                default:
                    break;
            }
        }
    }

    infile.close();

    bool choiceGood = false;
    do {
        cout << "\nWould you like to replace the current file or create a new
            file?\n";
        cout << "Enter 1 to replace the current file's contents.\n";
        cout << "Enter 2 to create a new file.\n";
        cout << "Choice: ";

        cin >> choice;

        if (isdigit(choice)) {
            if (atoi(&choice) > 0 && atoi(&choice) < 3)
                choiceGood = true;
            else
                cout << "\nInvalid Choice, Try again\n\n";
        } else
            cout << "\nInvalid Choice, Try again\n\n";

        cin.ignore(INT_MAX,'\n');

    } while (!choiceGood);

    if (choice == 2) {
        cout << "Enter the file name to access: ";
        cin >> fileToRead;
    }

    ofstream outfile;

    outfile.open(fileToRead.c_str());

    for (int i = 0; i < currentNumbers.size(); i++) {
        if (i == currentNumbers.size() - 1)
            outfile << currentNumbers[i];
        else
            outfile << currentNumbers[i] << " ";
    }
}

// This method allows input of a float

float Input::enterNumber()
{
    float current = 0;
    string currentString = "";

    cout << "\nEnter number: ";

    cin >> currentString;

    while (!stringToFloat.isStringAFloat(currentString)) {

        cout << "\nInvalid Value, try again\n\n";
        cout << "\nEnter number: ";

        cin >> currentString;
    }
```

```
        current = stringToFloat.getFloatValue();

        return current;
    }
```

```
// Name: James Small
// Program: 3B
// Class: CSE455
// Description: Class to convert string to float, if possible

#ifndef STRINGTOFLOAT_H
#define STRINGTOFLOAT_H

#include <string>

using namespace std;

class StringToFloat
{
    public:
        StringToFloat();
        bool isStringAFloat(string stringToTest);
        float getFloatValue();

    private:
        string currentString;
        float currentFloat;
};
#endif
```

```
// Name: James Small
// Program: 3B
// Class: CSE455
// Description: StringToFloat class implementation file

#include "StringToFloat.h"
#include <stdlib.h> // for atof
#include <ctype.h> // for isdigit

// Constructor which sets the currentFloat to 0

StringToFloat::StringToFloat()
{
    currentFloat = 0;
}

// This method takes a string and returns true or false if a float

bool StringToFloat::isStringAFloat(string stringToTest)
{
    currentString = stringToTest;
    int periodsCount = 0;
    bool nonDigitFound = false;
    bool isFloat = false;

    for (int i = 0;i < currentString.length(); i++) {
        if (!isdigit(currentString[i])) {
            if (currentString[i] == '.') {
                periodsCount++;
            }
            else if (currentString[i] == '-') {
                if (i != 0)
                    nonDigitFound = true;
            } else
                nonDigitFound = true;
        }
    }

    if (!nonDigitFound && periodsCount < 2) {
        isFloat = true;
        currentFloat = atof(currentString.c_str());
    }

    return isFloat;
}

// This method returns the float value

float StringToFloat::getFloatValue()
{
    return currentFloat;
}
```

Object Category Sizes in LOC per Method

Name: James Small

Program: 3B

Number: 5

Instructor: Dr. Concepcion

Language: C++

Object Size in LOC per Method (stddev method)

Type	V. Small	Small	Medium	Large	V. Large
Logic	0	0	0	0	0
I/O	24.6	24.6	24.6	24.6	24.6
Calc	0	0	0	0	0
Text	0	0	0	0	0
Data	0	0	0	0	0
Set-up	0	0	0	0	0

Object Size in LOC per Method (natural log method)

Type	V. Small	Small	Medium	Large	V. Large
Logic	0	0	0	0	0
I/O	24.6	24.6	24.6	24.6	24.6
Calc	0	0	0	0	0
Text	0	0	0	0	0
Data	0	0	0	0	0
Set-up	0	0	0	0	0

Compilation

```
jamess-imac:program AcousticTime$ g++ -c Input.cpp
jamess-imac:program AcousticTime$ g++ -c StringToFloat.cpp
jamess-imac:program AcousticTime$ g++ -o program3b program3b.cpp Input.o
StringToFloat.o
```

Test 1

```
jamess-imac:program AcousticTime$ ./program3b
What would you like to do?
Enter 1 to read from file.
Enter 2 to write to file.
Enter 3 to modify a file.
Enter 0 to quit.
Choice: a

Invalid Choice, Try again

What would you like to do?
Enter 1 to read from file.
Enter 2 to write to file.
Enter 3 to modify a file.
Enter 0 to quit.
Choice: 4

Invalid Choice, Try again

What would you like to do?
Enter 1 to read from file.
Enter 2 to write to file.
Enter 3 to modify a file.
Enter 0 to quit.
Choice:
ffadf

Invalid Choice, Try again

What would you like to do?
Enter 1 to read from file.
Enter 2 to write to file.
Enter 3 to modify a file.
Enter 0 to quit.
Choice: 1djfd
Enter the file name to access:
```

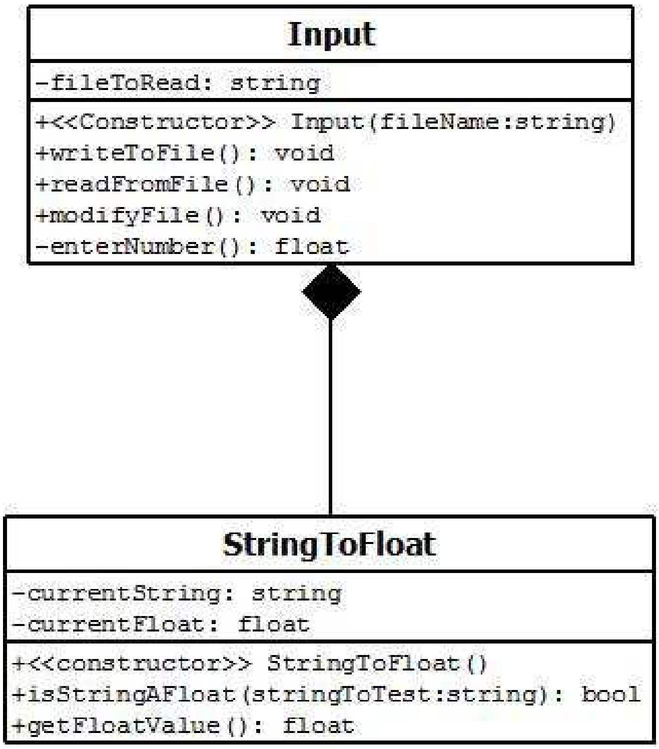
Test 2

```
jamess-imac:program AcousticTime$ ./program3b
What would you like to do?
Enter 1 to read from file.
Enter 2 to write to file.
Enter 3 to modify a file.
Enter 0 to quit.
Choice: 2
Enter the file name to access: test2
Enter the amount of numbers to write: a

Invalid number, Try again

Enter the amount of numbers to write: -1

Invalid number, Try again

Enter the amount of numbers to write: 4
Enter number 1: a

Invalid Value, try again

Enter number 1: dafd

Invalid Value, try again

Enter number 1: 5
Enter number 2:
```

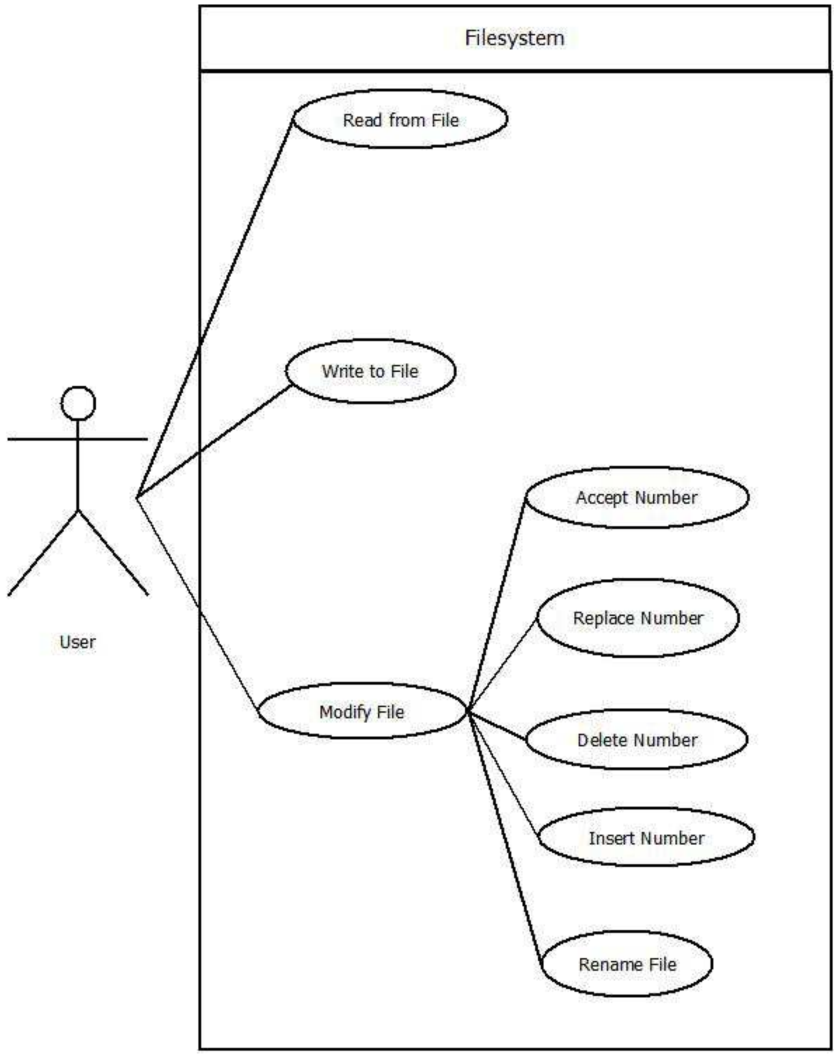
Test 3

```
jamess-imac:program AcousticTime$ ./program3b
What would you like to do?
Enter 1 to read from file.
Enter 2 to write to file.
Enter 3 to modify a file.
Enter 0 to quit.
Choice: 3
Enter the file name to access: james

What would you like to do with this number, 1?
Enter 1 to accept this number.
Enter 2 to replace this number.
Enter 3 to delete this number.
Enter 4 to insert a new number after current number.
Enter 5 to accept the remainder of the numbers.
Choice: a

Invalid Choice, Try again


What would you like to do with this number, 1?
Enter 1 to accept this number.
Enter 2 to replace this number.
Enter 3 to delete this number.
Enter 4 to insert a new number after current number.
Enter 5 to accept the remainder of the numbers.
Choice: bb

Invalid Choice, Try again


What would you like to do with this number, 1?
Enter 1 to accept this number.
Enter 2 to replace this number.
Enter 3 to delete this number.
Enter 4 to insert a new number after current number.
Enter 5 to accept the remainder of the numbers.
Choice: 7

Invalid Choice, Try again


What would you like to do with this number, 1?
Enter 1 to accept this number.
Enter 2 to replace this number.
Enter 3 to delete this number.
Enter 4 to insert a new number after current number.
Enter 5 to accept the remainder of the numbers.
Choice: 1

What would you like to do with this number, 3?
Enter 1 to accept this number.
Enter 2 to replace this number.
Enter 3 to delete this number.
Enter 4 to insert a new number after current number.
Enter 5 to accept the remainder of the numbers.
Choice:
```

UML Class Diagram



UML Use Case Diagram


Filesystem
Read from File
Write to File
User
Accept Number
Replace Number
Modify File
Delete Number
Insert Number
Rename File