PSP1.1 Project Plan Summary

Name: James Small
Program: 4B Number: 6
Instructor: Dr. Concepcion Language: C++

Summary	Plan	Actual	To Date
LOC/Hour	63.2	50.6	58.4
Planned Time	95		470
Actual Time		89	587
CPI (Cost-Performance Index)			0.801 (Planned/Actual)
% Reused	14	14.9	4.6
% New Reusable	12	25.3	12.1

Program Size (LOC)	Plan	Actual	To Date
Base (B)	216 (Measured)	216 (Measured)	
Deleted (D)	5 (Estimated)	4 (Counted)	
Modified (M)	5 (Estimated)	2 (Counted)	
Added (A)	95 (N-M)	73 (T-B+D-R)	
Reused (R)	50 (Estimated)	50 (Counted)	50
New and Changed (N)	100 (Estimated)	75 (A+M)	571
Total LOC (T)	356 (N+B-M-D+R)	335 (Measured)	1084
Total New Reusable	12	19	69

Time in Phase (min.)	Plan	Actual	To Date	To Date %
Planning	3	6	24	4.1
Design	9	11	59	10.1
Code	35	34	217	37

Compile	7	2	37	6.3
Test	30	15	172	29.3
Postmortem	11	21	78	13.3
Total	95	89	587	100

Defects Injected	**Actual**	**To Date**	**To Date %**
Planning	0	0	0
Design	0	1	5
Code	3	19	95
Compile	0	0	0
Test	0	0	0
Total Development	3	20	100

Defects Removed	**Actual**	**To Date**	**To Date %**
Planning	0	0	0
Design	0	0	0
Code	0	0	0
Compile	0	8	40
Test	3	12	60
Total Development	3	20	100
After Development			

Size Estimating Template

Name: James Small
Program: 4B Number: 6
Instructor: Dr. Concepcion Language: C++

BASE PROGRAM LOC	ESTIMATE	ACTUAL
BASE SIZE (B)	216	216
LOC DELETED (D)	5	4
LOC MODIFIED (M)	5	2

OBJECT LOC

BASE ADDITIONS	TYPE	METHODS	REL. SIZE	LOC	LOC
Input	I/O	6	Medium	99	228
TOTAL BASE ADDITIONS (BA)				99	228

NEW OBJECTS	TYPE	METHODS	REL. SIZE	LOC (New Reuse*)	
FileCheck	I/O	1	Small	12	19*
TOTAL NEW OBJECTS				12	19

REUSED OBJECTS

StringToFloat (3B)	50	50
REUSED TOTAL	50	50

PROBE Estimating Method:		SIZE C	TIME C
Estimated Object LOC (E):	E=BA+NO+M	116	
Regression Parameters:	β_0(size and time)	0	0
Regression Parameters:	β_1(size and time)	1.35254	1.91818
Estimated New and Changed LOC (N):	$N=\beta_0+\beta_1*E$	156.9	
Estimated Total LOC:	T=N+B-D-M+R	412.9	
Estimated Total New Reuse (sum of * LOC):		12	
Estimated Total Development Time:	$Time=\beta_0+\beta_1*E$		222.5
Prediction Range:	Range	20	20
Upper Prediction Interval:		176.9	242.5
Lower Prediction Interval:		136.9	202.5
Prediction Interval Percent:		N/A	N/A

Compilation

```
jamess-imac:program AcousticTime$ g++ -c Input.cpp
jamess-imac:program AcousticTime$ g++ -c StringToFloat.cpp
jamess-imac:program AcousticTime$ g++ -c FileCheck.cpp
jamess-imac:program AcousticTime$ g++ -o program4b program4b.cpp Input.o
StringToFloat.o FileCheck.o
```

Test 1

```
jamess-imac:program AcousticTime$ ./program4b
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: 1
Enter the file name to access: notthere

The filename doesn't exist
What would you like to enter a new filename?
Enter 1 to enter another filename.
Enter 0 to quit.
Choice: 1
Enter the file name to access: test1
1
2
3
4
5
jamess-imac:program AcousticTime$
```

Test 2

```
jamess-imac:program AcousticTime$ ./program4b
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

The filename already exists
What would you like to enter a new filename?
Enter 1 to enter another filename.
Enter 0 to quit.
Choice: 1
Enter the file name to access: test2b
Enter the amount of numbers to write: 5
Enter number 1: 1
```

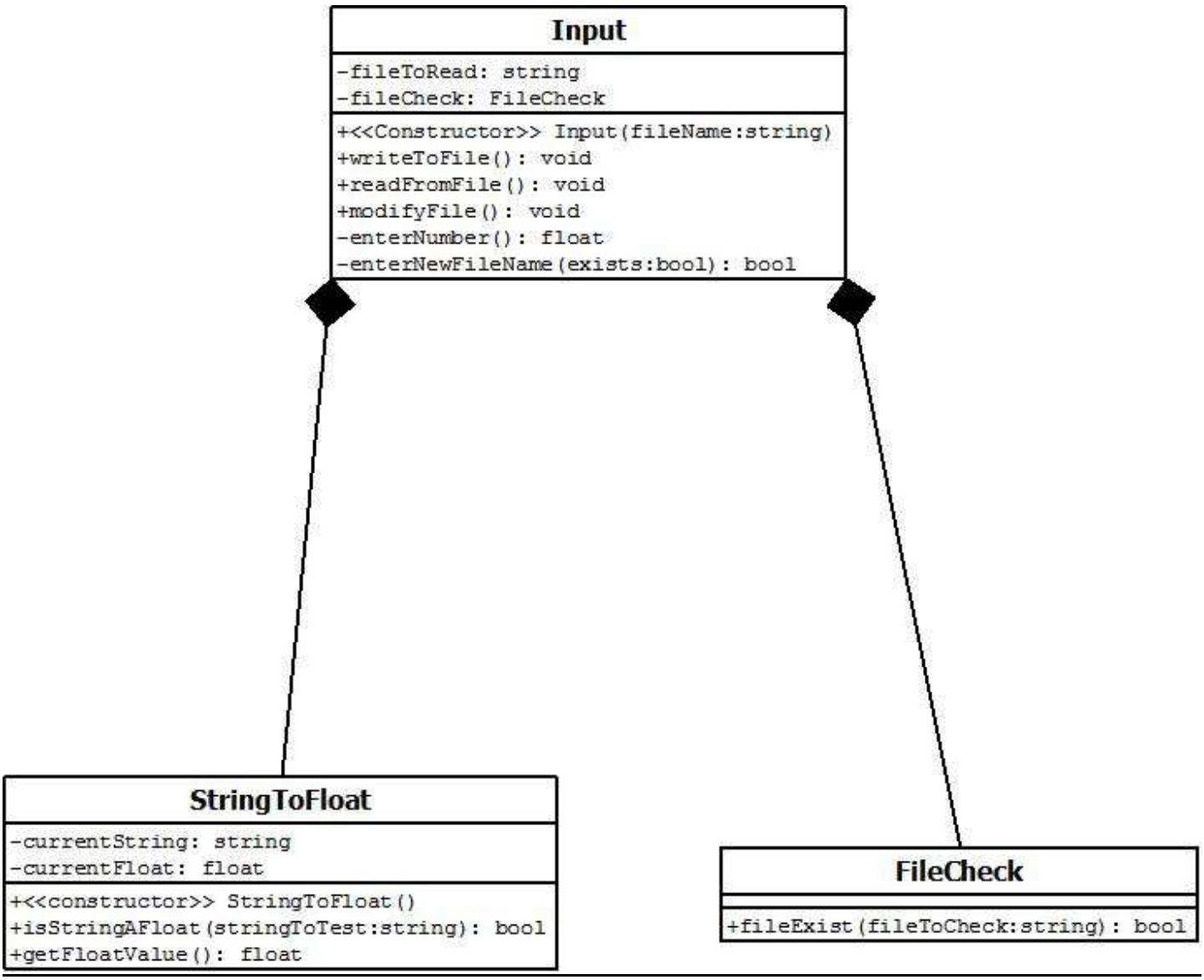
```
Enter number 2: 2
Enter number 3: 3
Enter number 4: 4
Enter number 5: 5
jamess-imac:program AcousticTime$
```

Test 3

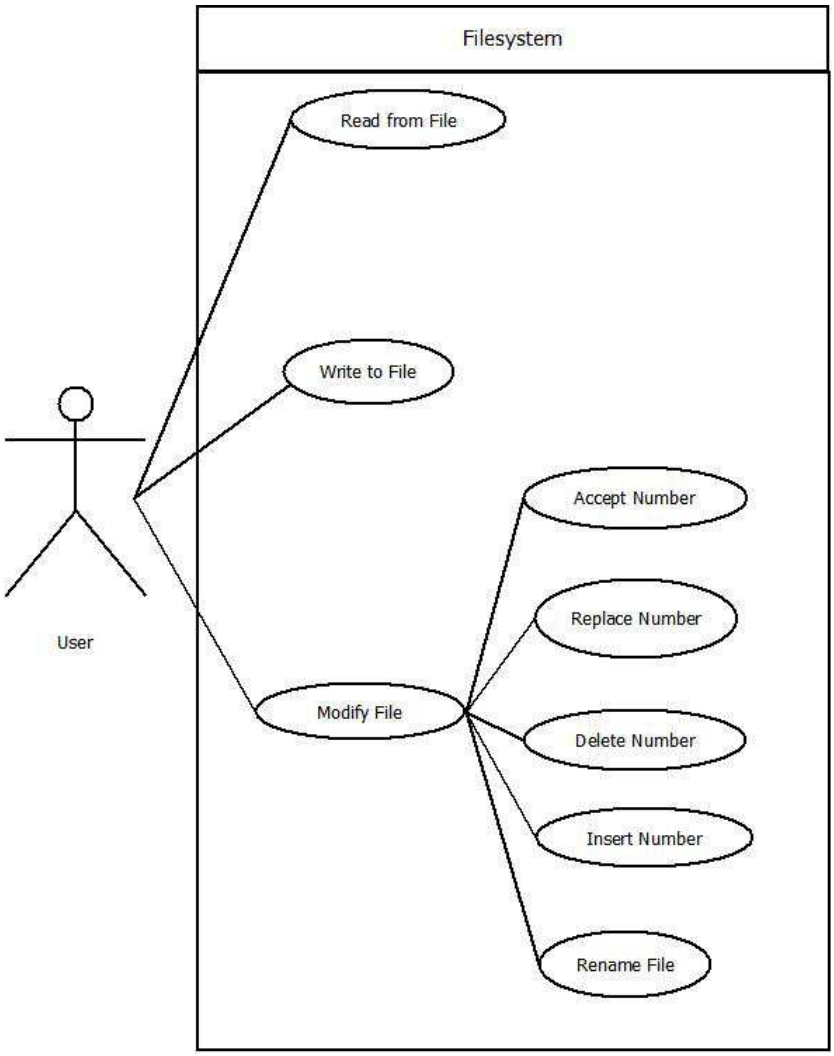
```
jamess-imac:program AcousticTime$ ./program4b
What would you like to do?
Enter 1 to read from file.
Enter 2 to write to file.
Enter 3 to modify a file.
Enter 0 to quit.
Choice: 3
Enter the file name to access: test3

The filename doesn't exist
What would you like to enter a new filename?
Enter 1 to enter another filename.
Enter 0 to quit.
Choice: 0
jamess-imac:program AcousticTime$
```

UML Class Diagram



UML Use Case Diagram



```
// Name: James Small
// Program: 4B
// 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') {

        Input input;

        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"
#include "FileCheck.h"

using namespace std;

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

private :
    string fileToRead;
    float enterNumber();
    bool enterNewFileName(bool exists);
    StringToFloat stringToFloat;
    FileCheck fileCheck;
};
#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()
{
    cout << "Enter the file name to access: ";
    cin >> fileToRead;
}

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

void Input::writeToFile()
{
    while (fileCheck.fileExist(fileToRead))
        if (!enterNewFileName(true))
            return;

    string count;
    float currentValue;
    string currentString = "";
    bool countGood = false;

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

        cin >> count;

        bool allDigitsInt = true;

        for (int i = 0; i < count.size(); i++)
            if (!isdigit(count[i]))
                allDigitsInt = false;

        if (allDigitsInt) {
            if (atoi(count.c_str()) > 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.c_str()); 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 == atoi(count.c_str()) - 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()
{
    while (!fileCheck.fileExist(fileToRead))
        if (!enterNewFileName(false))
            return;

    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()
{
    while (!fileCheck.fileExist(fileToRead))
        if (!enterNewFileName(false))
            return;

    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;

        while (fileCheck.fileExist(fileToRead))
            if (!enterNewFileName(false))
                return;
    }

    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;
}

// This method asks the user to enter a new filename
```

```
bool Input::enterNewFileName(bool exists)
{
    if (exists)
        cout << "\nThe filename already exists\n";
    else
        cout << "\nThe filename doesn't exist\n";

    char choice = 0;
    bool choiceGood = false;

    do {
        cout << "What would you like to enter a new filename?\n";
        cout << "Enter 1 to enter another filename.\n";
        cout << "Enter 0 to quit.\n";
        cout << "Choice: ";

        cin >> choice;

        if (isdigit(choice)) {
            if (atoi(&choice) >= 0 && atoi(&choice) < 2)
                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 == '1') {
        cout << "Enter the file name to access: ";
        cin >> this->fileToRead;
        return true;
    } else
        return false;
}
```

```
// Name: James Small
// Program: 4B
// Class: CSE455
// Description: Class to check if file exists in current
directory

#ifndef FILECHECK_H
#define FILECHECK_H

#include <string>

using namespace std;

class FileCheck
{
      public:
            bool fileExist(string fileToCheck);
};
#endif
```

```
// Name: James Small
// Program: 4B
// Class: CSE455
// Description: FileCheck class implementation file

#include "FileCheck.h"
#include <fstream>

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

bool FileCheck::fileExist(string fileToCheck)
{
    ifstream infile;

    infile.open(fileToCheck.c_str());

    infile.close();

    return infile;
}
```

```
// 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;
```

```
}
```