

University of Calgary, Schulich School of Engineering
ENSF 337: Programming Fundamentals
Final Exam – Dec 12, 2018
Exam Duration: 3 Hours
Exam Location: Aux Gym
Instructor: M. Moussavi

Your Last Name:
Your First Name:
Your Student Id:
Your Signature:

Section 1	/ 15
Section 2	/12
Section 3	/19
Section 4	/10
Section 5	/08
Section 6	/20
Section 7	/15
Section 8	/09
Section 9	/07
Total	/115

Please read these Instructions first:

- You may **not** use any electronic devices or calculator during the test.
- The test is **closed book**. You may **not** refer to books or notes during the test.
- Some problems are relatively **easy** and some are relatively **difficult**. Go after the easy marks first.
- To reduce distraction to other students, you are not allowed to leave during the last **15 minutes** of the test.
- Write all answers on the question paper and hand in the question paper when you are done.
- Please print your answers legibly. What cannot be read cannot be marked.
- When writing C++ code or program output, be clear and unambiguous about punctuation.
- Marks will be deducted for the following kinds of poor programming style: lack of clear, consistent indentation, unreasonable choice of variable names; code that is unnecessarily complicated, code that is inefficient.
- If you write anything that you do not want to be marked, put a large **X** through it and write “rough work” beside it.
- The official University of Calgary examination regulations are also printed on page 2.

EXAMINATION RULES AND REGULATIONS

STUDENT IDENTIFICATION

Each candidate must sign the Seating List confirming presence at the examination. All candidates for final examinations are required to place their University of Calgary I.D. cards on their desks for the duration of the examination. (Students writing mid-term tests can also be asked to provide identity proof.) Students without an I.D. card who can produce an acceptable alternative I.D., e.g., one with a printed name and photograph, are allowed to write the examination.

A student without acceptable I.D. will be required to complete an Identification Form. The form indicates that there is no guarantee that the examination paper will be graded if any discrepancies in identification are discovered after verification with the student's file. A Student who refuses to produce identification or who refuses to complete and sign the Identification Form is not permitted to write the examination.

EXAMINATION RULES

- (1) Students late in arriving will not normally be admitted after one-half hour of the examination time has passed.
- (2) No candidate will be permitted to leave the examination room until one-half hour has elapsed after the opening of the examination, nor during the last 15 minutes of the examination. All candidates remaining during the last 15 minutes of the examination period must remain at their desks until their papers have been collected by an invigilator.
- (3) All inquiries and requests must be addressed to supervisors only.
- (4) Candidates are strictly cautioned against:
 - (a) speaking to other candidates or communicating with them under any circumstances whatsoever;
 - (b) bringing into the examination room any textbook, notebook or memoranda not authorized by the examiner;
 - (c) making use of calculators and/or portable computing machines not authorized by the instructor;
 - (d) leaving answer papers exposed to view;
 - (e) attempting to read other student's examination papers.

The penalty for violation of these rules is suspension or expulsion or such other penalty as may be determined.

- (5) Candidates are requested to write on both sides of the page, unless the examiner has asked that the left hand page be reserved for rough drafts or calculations.
- (6) Discarded matter is to be struck out and not removed by mutilation of the examination answer book.
- (7) Candidates are cautioned against writing in their answer book any matter extraneous to the actual answering of the question set.
- (8) The candidate is to write his/her name on each answer book as directed and is to number each book.
- (9) A candidate must report to a supervisor before leaving the examination room.
- (10) Answer books must be handed to the supervisor-in-charge promptly when the signal is given. Failure to comply with this regulation will be cause for rejection of an answer paper.
- (11) If during the course of an examination a student becomes ill or receives word of a domestic affliction, the student should report at once to the supervisor, hand in the unfinished paper and request that it be cancelled. If physical and/or emotional ill health is the cause, the student must report at once to a physician/counsellor so that subsequent application for a deferred examination is supported by a completed Physician/Counsellor Statement form. Students can consult professionals at University Health Services or University Counselling Services during normal working hours or consult their physician/counsellor in the community.

Should a student write an examination, hand in the paper for marking, and later report extenuating circumstances to support a request for cancellation of the paper and for another examination, such a request will be denied.

- (12) Smoking during examinations is strictly prohibited.

RO 94-01

IMPORTANT NOTES:

For all of questions in this exam you can make the following assumptions:

- Any needed library header files are included, followed by: `using namespace std;`
- Size of `int` type is 4 bytes, size of `pointer` is 8 bytes, and size of `double` is 8 bytes.

Section 1 – Multiple Choice – (1 mark each)

1. What is the output of the following program:

<pre>void fun(int n){ if (n <= 0) return; cout << n + 1; fun (n-2); cout << n-1; }</pre>	<pre>int main() { fun (3); return 0; }</pre>
---	--

- a. 4020
 - b. 4010
 - c. 4001
 - d. 4102
 - e. 4110
 - f. None of the above
2. Consider the statement: `char x[8] = "ABC", y[] = "ABC";`
Which of the following is true in C/C++?
- a. `x == y`
 - b. `strcmp(x, y) != 0`
 - c. `sizeof(y) == sizeof(x)`
 - d. `strlen(y) < strlen(x)`
 - e. None of the above

3. Consider the following small program?

1	<code>static char s[4] = {0, 3, 4};</code>
2	<code>int main() {</code>
3	<code> s[0] = 'y';</code>
4	<code> s[0] = s[1] + s[2] + s[3];</code>
6	<code> return 0;</code>
7	<code>}</code>

- a. Program is erroneous because of line 1
 - b. Program is erroneous because of line 3;
 - c. Program is erroneous because of line 4;
 - d. both (b), and (c) are correct statements
 - e. None of the above
4. Which one of the following statements is **true**?
- a. A class constructor can be overloaded
 - b. A class constructor's return type is NOT void
 - c. All of the above is true
 - d. None of the above is true
5. Which one of the following statements is **false**?
- a. A class destructor can be overloaded
 - b. A class constructor's return type is NOT void
 - c. Statements (a) and (b) are both false
 - d. None of the above statements is false
6. Which line in the following code segment may create a compilation error?
- ```
char m1[] = "\0\0\0";
char *const m2 = &m1[0];
*m2 = 'X';
m2++;
```
- a. First line
  - b. Second line
  - c. Third line
  - d. Fourth line
  - e. None of the lines

7. Which one of the following statements is true:
- a. An argument in a C++ function can have a default value only if: All arguments to the right of that argument have a default value.
  - b. An argument in a C++ function can have a default value only if: All arguments to the left of that argument have a default value.
  - c. All of the above are correct.
  - d. None of the above is correct.
8. Which one of the following statements is true:
- a. A function in C++ can appear on the left-hand-side of an assignment operator, if its return type is void.
  - b. A function in C++ can appear on the left-hand-side of an assignment operator, if it doesn't have a return type.
  - c. A function in C++ can appear on the left-hand-side of an assignment operator, if its return type is a reference.
  - d. There is no way to call a function on the left-hand of an assignment operator.
9. Consider the following program:

|                                  |    |                          |
|----------------------------------|----|--------------------------|
| class Box {                      | 1  | int main(){              |
| public:                          | 2  | Box a(7777);             |
| Box(double x ) { this -> x = x;} | 3  | Box b = 7777;            |
| Box (): x(7777) {}               | 4  | Box c;                   |
| double getx() {return x;}        | 5  | Box* d;                  |
| private:                         | 6  | Box& e = a;              |
| double x, y;                     | 7  | Box*& f= d;              |
| };                               | 8  | // Assume more code here |
|                                  | 9  | return 0;                |
|                                  | 10 | }                        |

Which one of the statements in the given main function gives a compilation error?

- a. Line 3 in the the main
  - b. Line 4 in the main
  - c. Line 7 in the main
  - d. All of the above
  - e. None of the above
10. Consider the following C++ program and select the best answer:

|                                   |                      |
|-----------------------------------|----------------------|
| int x;                            | void fun() {         |
| int main() {                      | static int s = 5;    |
| for(int j = 0; j < 3; j++) fun(); | int m = 2;           |
| return 0;                         | s++;                 |
| }                                 | x++;                 |
|                                   | m++;                 |
|                                   | cout << x << s << m; |
|                                   | }                    |

- a. The output for some data will be garbage.
  - b. The output of the program is: 163273383
  - c. The output of the program is: 163163163
  - d. The output of the program is: 163263363
  - e. None of the above is true
11. Consider the following code segment, and select the best answer

```
vector<int> v1 = {11, 23, 99};
vector<int> v2 = {3, 9};
v2 = v1;
for(int i =0; i < 3; i++)
 cout << v2.at(i) << " ";
```

The output will be:

- a. 11 23 99
- b. 3 23 99
- c. 3 9
- d. 11 23 gargage
- e. There will be no output due to errounous statement(s) in this code segment.

12. Consider the following code segment and select the best answer:

```
string s1 = "863";
s1 += "79";
int y = s1.at(4) - s1.at(2);
```

- a. This code gives a compilation error on the second line.
- b. The value of y after this code segment will be 6
- c. The value of y after this code segment will be -6
- d. The value of y after this code segment will be 1
- e. None of the above

Consider the following C++ code segment and answer questions 14, 15, 16:

```
1 int main()
2 char *s1;
3 char s2[] = "ABA";
4 strcpy(s1, s2);
5 delete[] s1;
6 return 0;
7 }
```

13. There is a compilation error in this code

- a. true
- b. false

14. There is no logical error on line 4

- a. true
- b. false

15. There is no logical error on line 5

- a. true
- b. flase

16. What is the out put of the following code segment:

```
typedef vector<int> ROW;
vector<ROW> x(3, ROW(5)); // creating a vector of 3 ROW with 5 columns

// populating the matrix with integer numbers

for(int i = 1; i <= 3; i++)
 for (int j = 1; j <= 5; j++)
 x.at(i-1).at(j-1) = i+j;

for(int i = 0; i < 3; i++)
 for (int j = 0; j < 5; j++)
 if(i == j)
 cout << x[i].at(j);
```

- a. 246
- b. 035
- c. 136
- d. None of the above

## Section 2 – Short Answer Questions – 12 marks

**Part a (4 marks)** - A text file named `data.txt` contains following data:

```
2 527 126
```

What is the output of the following code segment assuming that text file `data.txt` is located in the same working directory and it is successfully accessed.

```
int a[4] = {10, 20, 30, 40};
char filename[100] = "data.txt";
ifstream in(filename);
if(!in){
 printf("File not found.");
 exit(1);
}
in >> a[0] >> a[1] >> a[2] >> a[3];
cout << a[0] << " " << a[1] << " " << a[2] << " " << a[3] << " ";
```

**Write your answer in the following space:**

**Part b** – Consider the following code segment in C++:

```
const char* x[3] = {"012345.987654321", "apple pie", "hello folks"};
ofstream out ("data.bin", ios::binary);
out.write(x[0], sizeof(*x));
cout << out.tellp();
out.close();
```

Now, answer the following questions assuming that operations of opening the file `data.bin` and writing into it have been successful.

**Question 1 (1 mark):** What would be the size of the file `data.bin` after this code segment?

**Question 2 (1 mark):** What characters are written into `data.bin`?

**Question 3 (3 marks):** in the following space write a for-loop that uses function `out.write` to write all three elements of array `x` (including backslash zero) into the binary file that is already open. Hint: you can use any other library function, if needed.

**Part c (3 marks)** – In the space beside the program and draw an AR diagram for point one assuming all memory allocations are successful.

```
void fun(int** x,int m){
 *x = new int [m];
}
int main(){
 int *p;
 p = new int[sizeof(int)];
 *p = 222;
 *(p + 1) = 100;
 fun(&p, 4);
 p[1] = 579;
 // point one
 return 0;
}
```

**Section 3: Class Activation Record and Program Output Determination (19 marks total)**  
Consider the following C++ program and answer the questions a, and b on the next page.

```
char m[] = "Bar";
class MyString {
 char *text;
 unsigned long length;
public:
 MyString(const char* s);
 MyString();
 ~MyString();
 MyString(const MyString& source);
 MyString (int n);
 void set_txt(const char* s);
 char* fun(const MyString& txt)const;
};

MyString::MyString(const char* s): length(strlen(s)), text(new char[strlen(s) + 1]){
 for(int i = 0; i <= length ; i++)
 text[i] = s[i];
 cout << "MyString ctor-1"<< endl;
}

MyString::MyString(int n):length(n), text(new char[n + 1]){
 for(int i = 0; i <= length ; i++)
 text[i] = 0;
 cout << "MyString ctor-2" << endl;
}

MyString::MyString():length(0), text(new char[1]){
 text[0] = '\\0';
 cout << "MyString Default Ctor"<< endl;
}

MyString::MyString(const MyString& source):length(source.length) {
 cout << "copy ctor" << endl;
 text = new char[length + 1];
 for(int i = 0; i <= length ; i++)
 text[i] = source.text[i];
}

MyString::~~MyString() {
 delete[] text;
 cout << "MyString dtor" << endl;
}

void MyString::set_txt(const char* s) {
 int i;
 for(i = 0; i < strlen(s) ; i++)
 text[i] = s[i];
 text[i] = 0;
}

char* MyString::fun(const MyString& txt)const{
 // point one
 return txt.text;
}

class Magazine {
public:
 Magazine(int i, int size): id(i), doc(size){}
 void fun(const MyString& txt);
 MyString& get() {return doc;}
private:
 int id;
 MyString doc;
};

void Magazine::fun(const MyString& txt){
 cout << txt.fun("TM") << endl;
}

int main() {
 Magazine* q= new Magazine(100, 4);
 m[0] = 'F';
 q -> get().set_txt(m);
 {
 Magazine b(*q);
 b.get().set_txt("B");
 b.fun(b.get());
 }
 delete q;
 return 0;
}
```

If you want you can detach this page, but you must write your student id on this page and return it back to us at the end of the exam. **Please don't write anything on this page.**

**Your Student ID Number:**

This page (page 8) is deliberately left blank. Students can use this page for their rough work, if needed. Material written on this page will not be marked. Please don't write your final answers on this page.



Question a (12 marks) - Draw a memory diagram for point one in the following space:

Question b (7 marks): What is the output of the program? Write your answer in the following space:

Section 4 (total of 10 marks) – Writing a member function for a class called Matrix

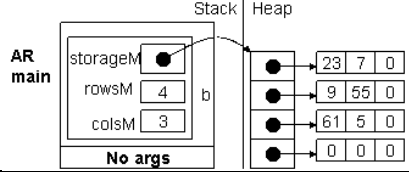
Consider the partial-definition of the class Matrix in the following box and answer the following question:

```
class Matrix {
public:
 Matrix(int n, int m);
 ~Matrix();
 void set(int row, int col, int value) {storageM[row][col] = value;}
 void resize (int new_row, int new_column);
 /* PROMISES: if new_row or new_column is less than or equal zero does nothing. Otherwise,
 * dynamically reallocates the space for storageM to resize the matrix to new_row by
 * new_column. If the new_row or new_column is less than this->rowsM or this->colsM the
 * data in resized matrix for the corresponding elements of the original matrix will be
 * lost. If the new_row or new_column is greater than this->rowsM or this->colsM the
 * the extra elements in the resized matrix will be all set to zero. */
private:
 int rowsM; // number of rows in a Matrix
 int colsM; // number of columns in a Matrix
 int ** storageM;
};

Matrix::Matrix(int r, int c): colsM(c),rowsM(r){
 storageM = new int* [rowsM];
 assert(storageM != nullptr);
 for(int i=0; i<r; i++){
 storageM[i] = new int[colsM];
 assert(storageM[i] != nullptr);
 for(int j=0; j < colsM; j++)
 storageM[i][j] = 0.0;
 }
}

int main(void) {
 Matrix b (4, 3); // creates a Matrix with 4 rows and 3 columns all initialized to zero
 b.set(0, 0, 23); // store 23 in the first row, first column
 b.set(0, 1, 7); // store 7 in the first row, second column
 b.set(1, 0, 9); // store 9 in second row, first column
 b.set(1, 1, 55); // store 55 in second row, second column
 b.set(2, 0, 61); // store 61 in third row, first column
 b.set(2, 1, 5); // store 5 in third row, second column
 // Point one
 return 0;
}
```

Here is the diagram that shows how object b appears on the memory at point one:



Question - Write the definition of member function `resize` in following space.

**Section 5: (8 marks) Writing class constructor**

Consider the partial definition of struct String and class StringArray that is used to create objects that contains an array of struct String. Then answer the question on the next page.

```
struct String {
 int lengthM; //the string length - number of characters excluding \0
 char* stringM; //a pointer to be used for dynamic allocation of an array of characters, with lengthM + 1
};

class StringArray {
public:

 StringArray(const char* arr[], int n);
 /* REQUIRES: n > 0 and arr points to an array of n elements that each element is a
 * char pointer pointing to a c-string.
 * PROMISES: creates an array of n String objects that storageM[i].stringM points
 * to a dynamically allocated memory on the heap, holding the copy of c-
 * string pointed by arr[i].
 */

 StringArray (const StringArray& src);

 ~StringArray();

 StringArray& operator = (const StringArray& v);

 int size() {return sizeM;}

private:

 String* storageM; // pointer to dynamically allocate space of an array of struct String object

 int sizeM; // number of elements in the StringArray
};

int main()
{
 const char* arr[3] = {"Hi", "Bye", "No"}; // Note: no space between last two quotation marks

 StringArray ms (arr, 3);

 // Point one

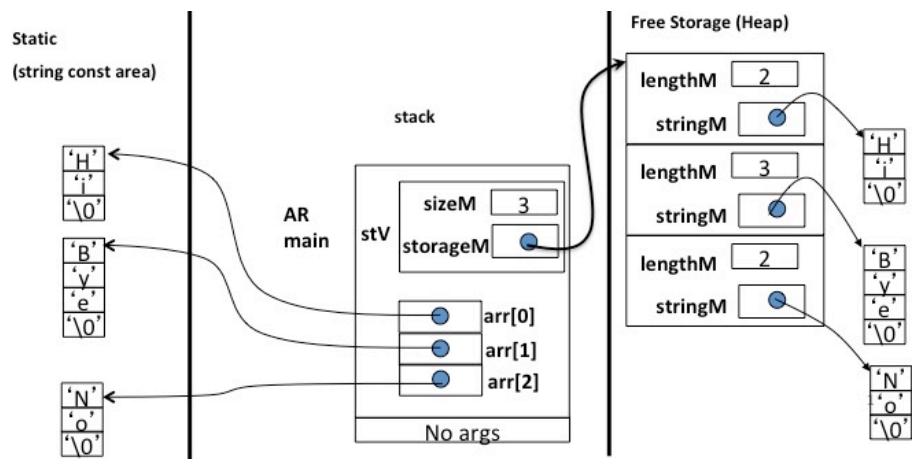
 return 0;
}
```

If you want you can detach this page, but you must write your student id on this page and return it back to us at the end of the exam. **Please don't write anything on this page.**

**Your Student ID Number:**

This page (page 11) is deliberately left blank. Students can use this page for their rough work, if needed. Material written on this page will not be marked. Please don't write your final answers on this page

**Question** – Please read the given function interface comment for the constructor of class StringArray and write its implementation in the following space. To help you to better understand how this constructor should work the AR diagram for point one in the main function is here:



Write the implementation of the constructor of class StringArray in this space:

**Section 6 - Linked List (20 marks)**

Consider the **partial** definition of a class called SimpleList and a class Node and answer the following questions:

```
class Node {
public:
 Node(int aItem): item(aItem), next(0) {}
 int item;
 Node *next;
};

class SimpleList {
public:
 SimpleList(): headM(0) {};
 void insert(const int& itemA);

 void insert_last(const int& itemA);
 /* PROMISE: inserts a node object with the itemA to the end of the list */

 ~SimpleList();
 SimpleList& operator= (const SimpleList& rhs);
private:
 Node *headM;
};

void SimpleList::insert(const int& itemA){
 Node *new_node = new Node(itemA);

 if (headM == 0 || itemA <= headM->item) {
 new_node->next = headM;
 headM = new_node;
 }
 else {
 Node *before = headM;
 Node *after = headM->next;

 while(after != 0 && itemA > after->item) {
 before = after;
 after = after->next;
 }
 new_node->next = after;
 before->next = new_node;
 // point ONE
 } // end of else
}

int main(){
 SimpleList the_list;
 the_list.insert(440);
 the_list.insert(220);
 the_list.insert(330);
 the_list.remove(550);
 return 0;
}
```

**Part a (7 marks)** Draw a memory diagram for point one, when reached for the **first time**.

**Part b (7 marks)** Write the definition of the **assignment operator** for class SimpleList, in the following space:

**Part c (4 marks)** Write the definition of function **insert\_last** that inserts a node to the end of the list:

**Part d (2 marks)** in the following space write the definition of the destructor for class SimpleList, using recursive technique. **No marks will be considered for non-recursive solutions:**

Section 7 - Drawing an AR Diagram for a Recursive Call

Part a (7 marks) - Draw a memory diagram when the value of **result** in the following program is **13**.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                             |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|
| <pre>#include &lt;iostream&gt; using namespace std;  int fun(int *a, int start, int end);  int main(void) {     int a[] = {5, 8, 1, 9, 11};      int result = fun(a, 0, 4);      cout &lt;&lt; "the result is: " &lt;&lt; result &lt;&lt; endl;      return 0; }  //-----  int fun(int *a, int start, int end) {     int Rval, Lval, result;      if(start == end)         return a[start];      else     {         int m = (start + end)/2;          Lval = fun(a, start, m);          Rval = fun(a, m + 1, end);     }      return result = Lval + Rval; }</pre> | <pre>// Draw your diagram in this box</pre> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|

Part b (2 marks) - What is the output of the program?

Part b (6 marks) In this section you should write the recursive solution of a function that returns the value of  $x^n$

```
int power(int x, int n);
/* REQUIRES: n >= 0
 * PROMISES: Returns x to the power of n. Example: when x is 2 n is 3 should return 8. */
```

Here is a sequence of numeric solutions that shows a recursive pattern that you should use in this solution.

$$x^{18} = x^{18/2} * x^{18/2} = x^9 * x^9$$
$$x^9 = x^{9/2} * x^{9/2} * x = x^4 * x^4 * x$$
$$x^4 = x^{4/2} * x^{4/2} = x^2 * x^2$$
$$x^2 = x^{2/2} * x^{2/2} = x * x$$
$$x^1 = x^0 * x^0 * x$$



## Section 8 (total of 9 marks)

In the area beside the following C++, draw a memory diagram for point one:

```
#include <iostream>
#include<cstring>
using namespace std;

char bar[] = {"ABCCD"};

void fun(char **z, int n){
 ((z + 2) - 2) = 'T';
 z[2][-3] = 'P';
 (*z)[1] = 'N';
 *z[1] = 'V';

 // point one
}

int main(void){
 char pc[2] = {'\0'};
 char *y[3];

 char *x = new char[sizeof(x)];
 strcpy(x, "Box");
 y[0] = bar + 2;
 y[1] = pc;
 y[2] = &x[3];
 fun(y, 3);
 return 0;
}
```

Section 9 (total of 7 marks)

In the following space write a C++ program that reads a text file with zero or more integer number, and creates an output text file, in which the numbers will be stored in tabular format with three columns and several rows, as needed. Here are a couple of examples of input files and expected output files:

| If input file contains                 | Output file will be (two spaces between numbers in each row) |
|----------------------------------------|--------------------------------------------------------------|
| 22 31 55 88 99 10<br>61 20 12 41<br>18 | 22 31 55<br>88 99 10<br>61 20 12<br>41 18                    |
| 22 31 55 88                            | 22 31 55<br>88                                               |

Note: The data in the output file can be separated by one or more spaces. It is up to you.

References to a few C++ file I/O member functions

`ofstream ofs(arg);`  
Opens a text file for output. *arg* must be a pointer to the beginning of a '\0'-terminated string; the string supplies the name of the file. Immediately following the attempt to open the file, `ofs.fail()` can be used to test for failure to open the file.

`ifstream ifs(arg);`  
Opens of file for input. *arg* must be a pointer to the beginning of a '\0'-terminated string; the string supplies the name of the file. Immediately following the attempt to open the file, `ifs.fail()` can be used to test for failure to open the file.

`get();` is a stream public member function that extracts characters from the stream, as unformatted input from a text file, and testing for failure ...

`eof();` is a stream public member function that returns the value `true` if the end of the input file has been reached;

`put(char c);` is a stream public member function that inserts character `c` into the stream.

`close();` is a stream public member function that closes the file currently associated the stream object

`fail();` is a stream public member function that returns `true` if either there is a logical error or read/write operations.