

COP3014-Foundations of Computer Science

Unit 10 Programming Assignment

Submit Assignment On Time!

Objectives:

1. Develop and use an ADT's to define a class to manage a dynamic array;
2. Use a default constructor to initialize the state of your class;
3. Use a destructor to de-allocate memory allocated using the new operator;

This assignment is an extension of Programming Assignment 5,(Unit 9's Programming Assignment). You will implement class called "**call_class**". The class will manage a dynamic array of call records called "*call_stats6.cpp*". **I have provide the driver "call_stats6.cpp" to help you implement this program.**

Your input data will be in the file "callstats_data.txt.

The descriptions of the functions you will implement are as follows:

1. **the default constructor to initialize the state of your class.** The default constructor will read the data from the file "callstats_data.txt" into the dynamic array call_DB. If call_DB becomes full, the function should call the function "double_size" to double the size (capacity) of call_DB. Remember, count and call_DB are private members of your class and do not need to be passed to an member function of your class.
2. **is_empty** is a Boolean public member function of the class. It has no formal parameter because count is a member of the state of the class (private member) and does not need to be passed to it because the state of the class is known to all member functions of the class. If count == 0 then true is returned; otherwise false is returned.
3. **is_full** is a Boolean public member function of the class. It has no formal parameters because count and size are members of the state of the class (private members) and do not need to be passed to it because the state of the class is known to all member functions of the class. If count == size then true is return; otherwise false. The size is the capacity which is the total number of cells allocated to call_DB.

4. **search** is an integer public member function that has only one formal parameter, the key. key is the cell phone number for the record you are search for. The array of records, call_DB and count are members of the state of the class and do not need to be passed to a member function of the class; The function will return the location of key in call_DB if it is there; otherwise -1 is returned.
5. **add** is a void public member function that inserts information for a call record into call_DB. Duplicates cell numbers are ok; add will prompt the user for the firstname, lastname, cell number, relays and call length. You may call process record to re-process when you add a new record. add has no formal parameters.
6. **remove** is a void public member function that deletes all records with the cell number stored in key. If duplicate records exist with the same cell number they must all be deleted. “remove” has only one formal parameter, the key.
7. **double_size** is a void public member function that doubles the capacity of call_DB. “double_size” has no formal parameters because size, count and call_DB are all members of the state of the class, call_class. First, size is multiplied by two; second, memory is allocated using “call_record *temp=new call_record[size]; third the records in call_DB are copied into temp with the statement “temp[i]=call_DB[i]” using a for loop. Forth, the old memory for call_DB is de-allocated using “delete [] call_DB”; Finally, call_DB is set to point to the new memory pointed to by temp using “call_DB = temp”.
8. **process** is a void public member function the has no formal parameter because call_DB and count are members of the state of the class.. The function *process will* calculate the net cost of a call (*net_cost*), the tax on a call (*call_tax*) and the total cost of the call (*total_cost*) using *the number of relay stations (relays)* and *the length in minutes of the call (call_length)* for all call records stored in call_DB. Please consider the following:
 - a. The *tax rate on a call (call_tax)* is simply based on the number of *relay stations (relays)* used to make the call (0<= *relays* <=5 then *call_tax* = 1%; 6<= *relays* <=11 then *call_tax* = 3%; 12<= *relays*<=20 then *call_tax* = 5%; 21<= *relays* <=50 then *call_tax* = 8%; *relays* >50 then *call_tax* =12%) .
 - b. The *net cost of a call* is calculated by the following formula: $net_cost = (relays / 50 \times 0.40 \times call_length)$.
 - c. *The tax on a call is equal to net_cost x call_tax / 100.*

- d. *The total cost of a call (rounded to the nearest hundredth)* is calculated by the following formula: $total_cost = net_cost + call_tax$. All tax and cost calculations should be rounded to the nearest hundredths.
9. **print** is a void public member function that has no formal parameters because count and call_DB are members of the state of the class. The function will print every field of every call_record in call_DB to the screen.
10. **the destructor** to de-allocate all memory allocated to call_DB. This function has no formal parameters because call_DB is a member of the state of the class. It will be called automatically by the compiler.

Use the driver “call_stats6.cpp” to help you implement this program.

Output Format for the Function *'print'*:

Consider the following sample output table when designing and implementing the function *'print'*. See section “**Format of Output**” below.

(The output is in the following order: firstname, lastname, cell phone number, relays, minutes, net cost, tax rate, call tax, total call cost)

Jean Hayward	9546321555	0	0	0	0.01	0	0
Marlon Brando	5612971340	5	50	2	0.01	0.02	2.02
John Kennedy	3051234567	8	25	1.6	0.03	0.05	1.65

Input Stream:

In the assignment you will declare one **ifstream** to bind your input to the file *"callstats_data.txt"*. Whenever a program performs *file i/o* you must include the **"fstream"** library.

Format of the input data file(input filename is "callstats_data.txt"): **Do not include column titles**

(The order of the columns are as follows: firstname, lastname, cell phone number, relays, minutes)

Jean Hayward	9546321555	0	0
Marlon Brando	5612971340	5	50
John Kennedy	3051234567	8	25
Hillary Clinton	7542346622	24	17
George Bush	3054432762	15	30
Barack Obama	9544321011	50	100
Donald Trump	8776219988	87	82
Bernie Sanders	9042224556	4	5
Harry Ford	7877176590	11	1
Michelle Obama	5617278899	20	45
Ann Dunham	9546321555	4	3
Vladimir Putin	5612971340	79	86
Harriet Tubman	3051234567	8	25
Oprah Winfrey	5611234444	24	118
Charlotte Ray	3054432762	115	25
Tina Turner	8776219988	265	22
Shirley Chisholm	9042224556	2	5
Maritza Correla	7877176590	89	67
Margaret Thatcher	5617278899	40	56

Format of Output to screen:

(the order of the columns is as follows: firstname, lastname, cell phone number, relays, minutes, net cost, tax rate, call tax, total call cost)

Jean Hayward	9546321555	0	0	0	0.01	0	0
Marlon Brando	5612971340	5	50	2	0.01	0.02	2.02
John Kennedy	3051234567	8	25	1.6	0.03	0.05	1.65
Hillary Clinton	7542346622	24	17	3.26	0.08	0.26	3.53
George Bush	3054432762	15	30	3.6	0.05	0.18	3.78
Barack Obama	9544321011	50	100	40	0.08	3.2	43.2
Donald Trump	8776219988	87	82	57.07	0.12	6.85	63.92
Bernie Sanders	9042224556	4	5	0.16	0.01	0	0.16
Harry Ford	7877176590	11	1	0.09	0.03	0	0.09
Michelle Obama	5617278899	20	45	7.2	0.05	0.36	7.56
Ann Dunham	9546321555	4	3	0.1	0.01	0	0.1
Vladimir Putin	5612971340	79	86	54.35	0.12	6.52	60.87

Harriet Tubman	3051234567	8	25	1.6	0.03	0.05	1.65
Oprah Winfrey	5611234444	24	118	22.66	0.08	1.81	24.47
Charlotte Ray	3054432762	115	25	23	0.12	2.76	25.76
Mary Carter	9544321011	43	10	3.44	0.08	0.28	3.72
Tina Turner	8776219988	265	22	46.64	0.12	5.6	52.24
Shirley Chisholm	9042224556	2	5	0.08	0.01	0	0.08
Maritza Correla	7877176590	89	67	47.7	0.12	5.72	53.43
Margaret Thatcher	5617278899	40	56	17.92	0.08	1.43	19.35

Handing in your program

Electronically submit *the source file "call_stats6.cpp"* in the Assignments area of blackboard before the due date and time. **Remember, to submit an assignment – even if it is late.....**