

CS 201 Homework 5

Solomon Himelbloom

November 2, 2020

- Repository Link: <https://github.com/techsolomon/cs201>
- Git Commits: <https://github.com/techsolomon/cs201/commits>
- This homework took approximately 4 hours to complete.

1 Design

Coming soon.

2 Post Mortem

Coming soon.

3 Answers to Questions

1. Written homework answers are coming soon.

4 Program 1

4.1 Sample Output/Screenshot

Listing 1: Sample Program Output

TBA

4.2 Git Commit Messages

Date	Message
2020-10-02	add: import hw4 files (bulls-and-cows, crud, fifo)
2020-11-01	add: hw5 (template CMakeLists.txt)
2020-10-28	rm: hw5/main
2020-10-28	add: fltk-trunc.cpp starting output
2020-10-28	add: hw5 (initial files)

4.3 Source Code

4.4 fltk-trunc.cpp

```
1 // fltk-trunc.cpp
2 // Solomon Himelbloom
3 // 28 October 2020
4 // FLTK library example for CS 201.
5
6 #include <FL/Fl.H>
7 #include <FL/Fl_Window.H>
8 #include <FL/Fl_Box.H>
9
10 int main(int argc, char **argv) {
11     Fl_Window *window = new Fl_Window(340,180);
12     Fl_Box *box = new Fl_Box(20,40,300,100,"Truncate + Quit");
13     box->box(FL_UP_BOX);
14     box->labelfont(FL_BOLD+FL_ITALIC);
15     box->labelsize(36);
16     box->labeltype(FL_SHADOW_LABEL);
17     window->end();
18     window->show(argc, argv);
19     return Fl::run();
20 }
```

5 Program 2

5.1 Sample Output/Screenshot

Listing 2: Sample Program Output

```
KEY: 1  
VALUE: 10  
KEY/VALUE PAIR STATUS: 110
```

5.2 Git Commit Messages

Date	Message
2020-10-12	add: key/value pairs to the database
2020-10-12	refactor: database to allow new records
2020-10-09	update: database import (map)
2020-10-09	add: ReadRecord to database.cpp
2020-10-09	add: UpdateRecord in database.cpp
rm: MyDatabaseRecord	
2020-10-09	add: bool logic to database.hpp
2020-10-09	add: MyDatabaseRecord struct in database.hpp
2020-10-09	add: crud.cpp and database.cpp template
2020-10-04	add: hw4 (initial files)

5.3 Source Code

5.4 crud.cpp

```
1 // crud.cpp
2 // Solomon Himelbloom
3 // 9 October 2020
4 // Database [crud source file] example for CS 201.
5
6 #include <iostream>
7 #include <stdio.h>
8 #include <string>
9 #include <map>
10
11 #include "database.hpp"
12
13 using std::cin;
14 using std::cout;
15 using std::endl;
16 using std::string;
17 using std::vector;
18
19 int main() {
20     std::string key = "1";
21     std::string value = "10";
22
23     cout << "KEY: " << key << endl;
24     cout << "VALUE: " << value << endl;
25     cout << "KEY/VALUE PAIR STATUS: " << key << value << endl;
26 }
```

5.5 database.cpp

```
1 // database.cpp
2 // Solomon Himelbloom
3 // 7 October 2020
4 // Database example for CS 201.
5
6 #include <iostream>
7 #include <stdio.h>
8 #include <string>
9 #include <map>
10
11 #include "database.hpp"
12
13 using std::cin;
14 using std::cout;
15 using std::endl;
16 using std::string;
17 using std::vector;
18
19 std::map<std::string, MyDatabaseRecord> theDatabase;
20
21
22 // Creates a new record within the database with a corresponding key value.
23 // bool CreateRecord(const std::string &key) {
24 //     auto it = theDatabase.find(key);
25 //     return true;
26 // }
```

```

27
28 // ReadRecord(key, record) copies the information from the database to
29 // a user supplied record
30 // @param {string} key
31 // @param {MyDatabaseRecord} record
32 // @returns false if the record does not exist
33 bool ReadRecord(const std::string &key, MyDatabaseRecord &record) {
34     auto it = theDatabase.find(key);
35     if (it == theDatabase.end()) {
36         return false;
37     }
38     // return = it->second;
39     return true;
40 }
41
42 // UpdateRecord(key, record) sets the database to the new value
43 // @param {string} key
44 // @param {MyDatabaseRecord} record
45 // @return true if operation successful
46 bool UpdateRecord(const std::string &key, const MyDatabaseRecord &record) {
47     auto it = theDatabase.find(key);
48     if (it == theDatabase.end()) {
49         return false;
50     }
51     theDatabase[key] = record;
52     return true;
53 }
54
55 // Deletes a record from the database given a key.
56 // bool DeleteRecord(const std::string &key) {
57 //     auto it = theDatabase.find(key);
58 // }
59
60 // Inputs a record into the database given a record.
61 // bool InputRecord(MyDatabaseRecord &record) {
62 //     auto it = theDatabase.find(key);
63 //     return true;
64 // }
65
66 // Prints a record from the database given a key.
67 // bool PrintRecord(const std::string &key) {
68 //     auto it = theDatabase.find(key);
69 //     cout << "Key: " << endl;
70 //     return true;
71 // }

```

5.6 database.hpp

```

1 // database.hpp
2 // Solomon Himelbloom
3 // 9 October 2020
4 // Database [header file] example for CS 201.
5
6 #ifndef DATABASE_HPP_
7 #define DATABASE_HPP_
8
9 #include <string>
10 #include <map>

```

```

11 struct MyDatabaseRecord {
12     // Replace this with information related to your database.
13     // std::string objectName{"obj"};
14     // std::string materialName{"mtl"};
15     // std::string diffuseColor{"diff"};
16     // std::string specularColor{"spec"};
17     // bool twoSided{false};
18
19     std::string key;
20     std::string value;
21 };
22
23 bool CreateRecord(const std::string &key);
24 bool ReadRecord(const std::string &key, MyDatabaseRecord &record);
25 bool UpdateRecord(const std::string &key, const MyDatabaseRecord &record);
26 bool DeleteRecord(const std::string &key);
27 bool InputRecord(MyDatabaseRecord &record);
28 bool PrintRecord(const std::string &key);
29
30 #endif /* DATABASE_HPP_ */

```

6 Program 3

6.1 Sample Output/Screenshot

Listing 3: Sample Program Output

```

Enter a four digit number (no repeats & negative to
quit): 1234
numberVector size: 4
guessVector size: 0
0 0 0 0
1 bull and 1 cows

```

6.2 Git Commit Messages

Date	Message
2020-10-12	refactor: database to allow new records
2020-10-12	add: if/else grammar logic in bulls-and-cows.cpp
2020-10-12	refactor: createNumberVector in bulls-and-cows.cpp
2020-10-09	add: if/else to bulls-and-cows.cpp
2020-10-09	add: randomNumber and if/else to bulls-and-cows.cpp
2020-10-09	add: guessingGame function to bulls-and-cows.cpp
2020-10-04	add: hw4 (initial files)

6.3 Source Code

```
1 // bulls-and-cows.cpp
2 // Solomon Himelbloom
3 // 7 October 2020
4 // Bulls and cows game example for CS 201.
5
6 #include <iostream>
7 #include <stdio.h>
8 #include <string>
9 #include <vector>
10
11 using std::cin;
12 using std::cout;
13 using std::endl;
14 using std::string;
15 using std::vector;
16
17 void randomNumber() {
18     int random_number = 1357;
19     cout << "Random integer: " << random_number << endl;
20     std::vector<int> createNumberVector { 1, 3, 5, 7 };
21     for (int& v : createNumberVector) {
22         std::cout << v << " ";
23     }
24     cout << endl;
25 }
26
27 void guessingGame() {
28     int bulls = 1;
29     int cows = 1;
30     vector<int> numberVector(4);
31     vector<char> guessVector(0);
32     cout << "numberVector size: " << numberVector.size() << endl;
33     cout << "guessVector size: " << guessVector.size() << endl;
34     for (auto i = 0; i < numberVector.size(); ++i) {
35         cout << numberVector[i] << " ";
36     }
37     cout << endl;
38     if (bulls != 1 && cows != 1) {
39         cout << bulls << " bulls and " << cows << " cows." << endl;
40     }
41     else if (bulls != 1 && cows == 1) {
42         cout << bulls << " bulls and " << cows << " cow." << endl;
43     }
44     else if (bulls == 1 && cows == 1) {
45         cout << bulls << " bull and " << cows << " cows" << endl;
46     }
47     else {
48         cout << bulls << " bull and " << cows << " cow" << endl;
49     }
50 }
51
52 int main(int argc, char *argv[]) {
53     int user_input = 0;
```



```

65     int random_number = 1357;
66
67     cout << "Enter a four digit number (no repeats & negative to quit): ";
68     cin >> user_input;
69
70     if (user_input == random_number) {
71         randomNumber();
72     }
73
74     else if (user_input > 9999) {
75         cout << "Input must be 4 digits in length (less than 10,000)." << endl;
76     }
77
78     else if (0 < user_input && user_input < 1000) {
79         cout << "Input must be 4 digits in length (greater than 999)." << endl;
80     }
81
82     else if (user_input < 0) {
83         randomNumber();
84     }
85
86     else {
87         guessingGame();
88     }
89
90     return 0;
91 }

```

7 Program 4

7.1 Sample Output/Screenshot

Listing 4: Sample Program Output

Hello, FIFO/LIFO.

7.2 Git Commit Messages

Date	Message
2020-10-09	update: fifo-lifo.cpp comments
2020-10-09	add: Fifo, Lifo, Push, Pop function templates
2020-10-09	add: fifo-lifo.cpp template file
2020-10-04	add: hw4 (initial files)

7.3 Source Code

```
1 // fifo-lifo.cpp
2 // Solomon Himelbloom
3 // 9 October 2020
4 // FIFO/LIFO example for CS 201.
5
6 #include <iostream>
7 #include <stdio.h>
8 #include <string>
9 #include <vector>
10
11 using std::cin;
12 using std::cout;
13 using std::endl;
14 using std::string;
15 using std::vector;
16
17 // First-in First-Out
18 void FifoPush(vector<string> &container, const string &item);
19 void FifoPop(vector<string> &container, string &item);
20
21 // Last-In First-Out
22 void LifoPush(vector<string> &container, const string &item);
23 void LifoPop(vector<string> &container, string &item);
24
25 // Shared functionality
26 bool IsContainerEmpty(const vector<string> &container);
27 void PrintContainer(const vector<string> &container);
28
29 // Implement these two tests to verify your functions work
30 // with at least the sequence:
31 //     push "A", push "B", push "C", push "D"
32 //     push   , pop   , pop   , pop
33 bool TestFifo();
34 bool TestLifo();
35
36 int main() {
37     cout << "Hello, FIFO/LIFO." << endl;
38 }
```
