

```
1  using namespace std;
2
3  class ui {
4  private:
5      void errorMessage(string);
6  public:
7      const string SCHOOLNAME = "University of Colorado Colorado Springs";
8      const string TAB = "    ";
9      void genHeader();
10     void genMainMenu();
11     void clearConsole();
12     string askStudentName();
13     int askOperation();
14     int askLevel();
15     void askQuestion(int, int, int, bool);
16     int getAnswer();
17     void echoStudentScores(progress); // progress[] & );
18     void answerEchoResult(bool);
19     void answerChoiceNotVaild();
20 };
21
22 void ui::errorMessage(string message) {
23     // Helper function that allows us to produce consistant error messages
24     // Requirements: N/A
25     cout << "An error has occured in the program." << endl;
26     cout << "The error condition reported: " << message << endl;
27     cout << "You can try that action again, but you may get the same result :(";
28     cout << endl;
29     cout << endl;
30 }
31
32 void ui::genHeader() {
33     // Generates a welcome and instructions for the
34     // user
35     // Requirements: N/A
36
37     cout << "Welcome to the Computer Aided Instruction System" << endl;
38     cout << "This software licensed to: " << SCHOOLNAME << endl;
39     cout << endl;
40     cout << endl;
41 }
42 void ui::genMainMenu() {
43     // Generates the main menu
44     // Requirements: N/A
45     cout << "Please enter your menu choice: " << endl;
46     cout << TAB << "1 - Answer questions" << endl;
47     cout << TAB << "2 - View Student scores" << endl;
48     cout << TAB << "0 - exit" << endl;
49 }
50
51 void ui::clearConsole() {
52     // Clears the console using a while loop and endl
53     // Requirements: N/A
```

```
54     for (int i = 0; i < 80; i++) {
55         cout << endl;
56     }
57 }
58 string ui::askStudentName() {
59     // Asks for the student's name and returns it in the
60     // form "<first name> <last name>"
61     // Requirements: 330
62     string first; // First Name
63     string last;  // Last Name
64
65     // Ask for the names
66     cout << "Please enter your first name: ";
67     cin >> first;
68     cout << "Please enter your last name: ";
69     cin >> last;
70
71     // Combine and return
72     return first + " " + last;
73 }
74
75 int ui::askOperation() {
76     // Asks which operation the student would like to do
77     // Reuquirements: N/A
78     int tempAnswer; // The answer given by the user
79
80     // Output the options to the screen
81     cout << "Please enter the operation you would like to do:" << endl;
82     cout << TAB << "1 - Addition" << endl;
83     cout << TAB << "2 - Subtraction" << endl;
84     cout << TAB << "3 - Multiplication" << endl;
85     cout << TAB << "4 - Division" << endl;
86     cout << TAB << "5 - A mixture of all four" << endl;
87
88     // Get the user's answer
89     tempAnswer = getAnswer();
90
91     // Some error checking
92     while (tempAnswer < 1 || tempAnswer > 5) {
93         errorMessage("That is an invalid menu choice.");
94         tempAnswer = getAnswer();
95     }
96
97     // Return the user's choice
98     return tempAnswer;
99 }
100
101 int ui::askLevel() {
102     // Asks the user which level the want to do
103
104     int tempAnswer; // The answer give by the user
105
106     // Output the option to the screen
```

```
107     cout << "Please enter which level you would like to try (1-3)" << endl;
108
109     // Get the user's answer
110     tempAnswer = getAnswer();
111
112     // Some error checking
113     while (tempAnswer < 1 || tempAnswer > 3) {
114         errorMessage("That is an invalid menu choice.");
115         tempAnswer = getAnswer();
116     }
117
118     // Return their choice
119     return tempAnswer;
120 }
121
122 void ui::askQuestion(int numOne, int numTwo, int op, bool secondTry = false) {
123     // Asks the student a question
124     // Inputs: first number, second number, second attempt
125     // Requirements: 120
126     string textOperator; // A textual representation of the operator
127
128     // Populate the textOperator field
129     switch (op) {
130     case 1:
131         textOperator = " + ";
132         break;
133     case 2:
134         textOperator = " - ";
135         break;
136     case 3:
137         textOperator = " * ";
138         break;
139     case 4:
140         textOperator = " / ";
141         break;
142     default:
143         textOperator = " ? ";
144         break;
145     }
146
147     // If this is the user's second try, output a notation to that effect
148     if (secondTry) {
149         cout << "(Second attempt) ";
150     }
151
152     // Output the actual problem
153     cout << numOne << textOperator << numTwo << " = " << endl;
154 }
155
156
157 int ui::getAnswer() {
158     // Gets the answer from a student using cin
159     // Returns the answer given
```

```

160     // Requirements: 130
161     int temp; // The answer given
162
163     cout << "Enter your answer: ";
164
165     cin >> temp;
166
167     return temp;
168 }
169 void ui::echoStudentScores(progress studentProgress) {
170     // Outputs the student scores from each item of the
171     // array to the screen
172     // Requirements: 220, 230, 240, 330, 340
173     cout << "Displaying scores for: " << studentProgress.getStudentName() << endl;
174     cout << "Number correct: " << studentProgress.getNumTrue() << endl;
175     cout << "Number wrong: " << studentProgress.getNumFalse() << endl;
176
177     // Let the user know if they're ready to go on to the next level
178     if ((studentProgress.getNumTrue() /
179         (studentProgress.getNumTrue() + studentProgress.getNumFalse()))
180         > 75 / 100) {
181         cout << studentProgress.getStudentName() << " is ready to go on to the"
182             << " next level" << endl;
183     }
184
185     // new line, for luck
186     cout << endl;
187 }
188 void ui::answerEchoResult(bool resultStatus) {
189     // Outputs the text indicating the result to the user
190     // Input: Whether the answer is true or false
191     // Requirements: 150, 160, 170, 180
192     int number;
193
194     string right[4] = { "Very good!", "Excellent!", "Nice work!",
195         "Keep up the good work!" }; // Collection of phrases for right
196     string wrong[4] = { "No. Please try again!", "Wrong. Try once more.",
197         "Don't give up!", "No. Keep trying." }; // Collection of phrases for wrong
198
199     // Random number for which one to use
200     number = rand() % 4;
201
202     // Output to the screen
203     if (resultStatus) {
204         cout << right[number] << endl;
205     }
206     else {
207         cout << wrong[number] << endl;
208     }
209 }
210
211
212 void ui::answerChoiceNotVaild() {

```

```
213      // Outputs an error message to the user.  
214      // Requirements: N/A  
215      errorMessage("That menu choice is not valid ");  
216  }  
217
```