

MINI PROJECT

```
#include <iostream>

#include <string>

#include <vector> // We use vector to easily hold our list of questions


// Use std namespace to avoid typing std:: constantly
using namespace std;


/*
 * This is the main blueprint for our questions.
 * This directly relates to your experiments #1 (Student class),
 * #2 (Complex class), and #4 (Rectangle class).
 *
 * A 'class' bundles data (attributes) and functions (methods)
 * that work on that data.
 */
class QuizQuestion {
// 'private' means these variables can ONLY be accessed
// by functions that are members of this class.
private:
    string questionText;
    string options[4]; // Array to hold 4 options
    int correctOption; // The index (0-3) of the correct answer


// 'public' means these functions can be called from
// outside the class (like in our 'main' function).
```

public:

```
// This is a 'constructor'. It's a special function
```

```
// called when you create a new object.
```

```
// This relates to your experiment #4!
```

```
QuizQuestion(string q, string o1, string o2, string o3, string o4, int correct) {
```

```
    questionText = q;
```

```
    options[0] = o1;
```

```
    options[1] = o2;
```

```
    options[2] = o3;
```

```
    options[3] = o4;
```

```
    correctOption = correct;
```

```
}
```

```
// This is a 'member function'. It's a function
```

```
// that "belongs" to the class.
```

```
// This is just like the 'display' or 'add' functions
```

```
// in your experiments.
```

```
//
```

```
// It returns 'true' if the user was correct, 'false' otherwise.
```

```
bool ask() {
```

```
    cout << "-----\n";
```

```
    cout << questionText << endl;
```

```
// Loop to display all 4 options
```

```
for (int i = 0; i < 4; i++) {
```

```
    cout << i + 1 << ". " << options[i] << endl;
```

```
}
```

```

int userAnswer;

cout << "Enter your choice (1-4): ";

cin >> userAnswer;


// Check the answer
// (userAnswer - 1) converts from 1-4 to 0-3
if (userAnswer - 1 == correctOption) {
    cout << "Correct!" << endl;
    return true;
} else {
    // FIX: This now correctly prints the number (e.g., "3.") and the text (e.g., "Paris")
    cout << "Wrong. The correct answer was " << (correctOption + 1) << ". " <<
options[correctOption] << endl;
    return false;
}
}

}; // Don't forget the semicolon at the end of a class!

```

```

// This is where your program starts

int main() {

    cout << "\n";

    cout << " Welcome to Brainquest! \n";

    cout << "\n\n";


// 2. Create your questions using the class constructor

// We are creating 'objects' (or 'instances') of our 'QuizQuestion' class.

QuizQuestion q1("What is the capital of France?",

```

```
"Berlin", "Madrid", "Paris", "Rome", 2);
```

```
QuizQuestion q2("What is 2 + 2?",  
    "3", "4", "5", "6", 1);
```

```
QuizQuestion q3("What C++ keyword is used to create a 'blueprint'?",  
    "struct", "object", "main", "class", 3);
```

```
// Add your questions to a simple C-style array.
```

```
// This is an easier alternative to a vector.
```

```
// We must define the size as a constant.
```

```
const int numQuestions = 3;
```

```
// Create the array and initialize it with your question objects
```

```
QuizQuestion questionsList[numQuestions] = {q1, q2, q3};
```

```
// To add more questions:
```

```
// 1. Create a q4, q5, etc.
```

```
// 2. Change numQuestions to the new total
```

```
// 3. Add q4, q5 to the list in {} above
```

```
int score = 0; // Variable to keep score
```

```
// 5. Use a 'for' loop to ask all questions
```

```
// We loop 'numQuestions' times
```

```
for (int i = 0; i < numQuestions; i++) {
```

```
    // Call the 'ask' member function on each question object
```

```
    if (questionsList[i].ask() == true) {
```

```
        score++; // Add to the score if the function returned true
    }
}

// 6. Show the final score
cout << "\n=====\\n";
cout << "Game Over! Your final score is: " << score << "/" << numQuestions << endl;
cout << "=====\\n";

return 0;
}
```