Here's what's happening with the input buffer when you run your program:

```
std::cin >> name;
```

You're waiting for input. When you enter "Ryan Cleary", and press enter, the input buffer contains:

```
Ryan Cleary\n
```

Now your cin reads input as normal, stopping at whitespace, leaving your buffer like this:

```
 Cleary\n
```

Note the beginning space, as it stops after reading Ryan. Your first variable now contains Ryan. If, however, you want the full name, use std::getline. It will read until a newline, not just whitespace. Anyway, continuing on:

```
std::cin >> age;
```

Now you're getting another input. There's already something there, though. It skips the whitespace until it can start reading, leaving the buffer with just:

```
\n
```

Your second variable gets the text Cleary. Note the newline still in the buffer, which brings me to the second part. Replacing system ("pause"); in a way that always works is tricky. Your best bet is usually to live with a less-than-perfect solution, or as I like to do, one that isn't guaranteed to do exactly what it says:

```
std::cin.get(); //this consumes the left over newline and exits without waiting
```

Okay, so cin.get() didn't work. How about this:

```
std::cin.get(); //consume left over newline
std::cin.get(); //wait
```

That works perfectly, but what if you copy-paste it somewhere where the newline isn't left over? You'll have to hit enter twice!

The solution is to clear the newline (and anything else) out, and then wait. This is the purpose of cin.sync(). However, as seen in the notes section, it is not guaranteed to clear the buffer out like it says, so if your compiler chooses not to, it can't be used. For me, however, it does exactly that, leaving a solution of:

```
std::cin.sync(); //clear buffer
std::cin.get(); //wait
```

The main bad thing about system("pause"); is that you have no idea what program it will run on someone else's computer. They could've changed pause.exe or put one that's found first, and you have no way of knowing. This could potentially ruin their computer due to it being possibly *any* program.
https://stackoverflow.com/questions/11462021/issue-with-cin-when-spaces-are-inputted-using-string-class