Lab: Pointer Operators

Tutorial Lab: Reference and Dereference Operators

Pointers are used to store the memory address of a particular variable. Two important operators associated with pointers are the **reference** `&` and the **dereference** operators `*`. The `&` operator returns the **memory address** of a variable and the `*` operator returns the **value** or content of the variable being pointed to.

```
bool b = true;
bool* p = &b;

cout << p << endl; //prints b's memory address
cout << boolalpha << *p << endl; //prints b's value
```

A pointer can also point to **another pointer**. When doing so, the new pointer will be denoted with *two* asterisk symbols `**`. `**` is also used to dereference a pointer twice.

```
bool b = true;
bool* p = &b;
bool** p2 = &p; //p2 points to p

cout << p2 << endl; //prints p's memory address
cout << *p2 << endl; //prints p's content which is b's address
cout << boolalpha << **p2 << endl;
//p2 is dereferenced twice to print b's value
```

Lab Challenge: Pointer Keys

Pointer Keys

You are trying to come up with a set of pointers or *keys* that, when referred to, will be able to tell you the age of each of your family members. For example, the pointer `amy` should be associated with the variable `age1`.

So far you have the following information in your code:

```
#include <iostream>
using namespace std;

int main() {

  int age1 = 12;
  int age2 = 31;
  int age3 = 29;
  int* amy;
  int* bob;
  int** carol;

  //add code below this line


  cout << *amy << endl; //do not edit

  cout << *bob << endl; //do not edit

  cout << **carol << endl; //do not edit


  //add code above this line

  return 0;

}
```

Pointer Challenge

challenge

Assignment:

Your task is to associate the pointers within the code to their respective variables. The output of your program should produce:

```
Amy's age is: 12
Bob's age is: 31
Carol's age is: 29
```

Requirement:

To receive credit for the challenge, you need to do the following:
* Only **add** to the existing code, **do not** make any changes, otherwise, you will not receive credit for your work. If you accidentally delete any existing code, you can copy the original code shown above back into your program.

- You **must** use the pointers `amy`, `bob`, and `carol` in your code and assign the pointers to their appropriate data.
- **NOTE** that the pointer `carol` is a pointer to another pointer. You will need to create a new pointer in your code to make the association.