

9 September 2021

1. **Warm up:** Answer the following True / False questions.

- (a) Several pointers can point to the same `int`
- (b) One pointer can point to several `ints`
- (c) It is possible to create an array that holds both `int` and `char` types.
- (d) It is possible to create an array that holds pointers to either both `int` or `char` types.

The next two problems refer to the following uncompiled C++ files.

```
arrows.cpp
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int num = 10;
5      int* ptr = &num;
6      int& ref = *ptr;
7      cout << ref << endl;
8      return 0;
9  }
```

```
database.cpp
#include <iostream>
#include <string>
using namespace std;
using std::string;
#include "header.h"
int main() {
    Bank Trusty;
    Employee Dusty;
    ...
    return 0;
}
```

```
header.h
struct Bank {
    string name;
    int employeeNum;
    Bank();
};
Bank::Bank() {employeeNum = 0;}

class Employee {
    string name;
    ...
    Bank employer;
public:
    Employee();
};
Employee::Employee() {}
```

1. This question is about the program that results from compiling `arrows.cpp`.

- (a) What is the output of the program?
- (b) Change line 7 so that the output is the address of `num`.
- (c) What would happen (if anything) if the reference `ref` instead of the pointer `ptr` was defined first? That is, what will happen if lines 5 and 6 are replaced with:

```
5      int& ref = *num;
6      int* ptr = &ref;
```

- (d) Suppose lines 6 and 7 are changed to

```
6      int** newptr = &ptr;
7      cout << newptr << endl;
```

What will be the output? How can you change this line 7 so that the output is 10?

2. This question is about `database.cpp` and `header.h`.

- (a) For each the following options to place in the line `...` of `database.cpp`, decide whether or not an error will be produced when the file is compiled. If no error is produced, what will be the output when the resulting program is executed?
 - i. `cout << Trusty.employeeNum << endl;`
 - ii. `cout << Trusty.name << endl;`
 - iii. `Dusty.name = "Gutsy";`
 - iv. `Trusty.name = "Musty"; cout << &Trusty.name << endl;`
 - v. `Dusty.employer = Trusty; Dusty.employer.employeeNum++;`
- (b) Create new public functions `setName` and `getName`, with constructors, for the class `Employee` that set the string `name` and that return it, respectively.