

Name: _____

Solutions

1. (1 pt.)

- **Read all material carefully.**
- *If in doubt whether something is allowed, ask, don't assume.*
- You may refer to your books, papers, and notes during this test.
- E-books may be used *subject to the restrictions* noted in class.
- Computers are not permitted, except when used strictly as e-books or for viewing ones own notes.
- Network access of any kind (cell, voice, text, data, ...) is not permitted.
- Write, and draw, carefully. Ambiguous or cryptic answers receive zero credit.
- Use class and textbook conventions for notation, algorithmic options, etc.

Write your name in the space provided above.

2. (4 pts.) Find at least one significant error in the following C++ program. Explain why it is an error and how best it may be fixed.

```
1  const int x = 5;
2  cin >> y;
3  if (y = 3) x *= x**2;
```

Ⓐ [There are multiple errors; any one suffices for full credit.]

- (1) The variable `x` is declared as a `const int` which means it cannot be assigned to after the initialization on the first line, but the last line assigns to it.
 - (2) On the last line, a nonexistent (in C++) operator `**` is used.
 - (3) Also on the last line, the predicate following the `if` keyword uses an assignment (`=`) instead of an equality boolean operator (`==`). This is not technically a C++ error, but is likely to be a programming/logic error because, as a predicate, that assignment will always result in true (due to its value being 3 which is nonzero and hence mapped to true).
3. (5 pts.) Provide a *single C++ statement* that will read, from *standard input*, appropriate values into the variables `x`, `y`, and `z` (in that order) where the variables are defined as follows:

```
1  int x, z;
2  float y;
```

Ⓐ `cin >> x >> y >> z;`

4. (15 pts.) Provide **well-formatted source code of a complete C++ program** that:
1. Prints a prompt "First name: " to *standard error* (note, not standard output).
 2. Waits for a response on *standard input*, terminated by a newline.
 3. Stores what is read as a string-valued first name in a suitably defined variable. (Note that the name may contain spaces, for names such as Mary Jane.)
 4. Repeats the above three steps for the middle name and then again for the last name (modifying the prompt and variable name suitably).
 5. Writes the following to **standard output**, with `fName` `mName` and `lName` replaced by the first, middle, and last names as read earlier:
`Hello, fName mName lName!`

Explain your answer briefly, especially to qualify for partial credit.

Ⓐ The code here uses the **string** type for the three variables and then reads each using **getline** in order to permit reading values with spaces. The rest of the program is about printing prompts and the result to the proper streams in the usual way. The standard headers **iostream** and **string** are needed for using the standard IO streams and string types, respectively. The **using namespace** line allows the use of **cin**, **cout**, and **cerr** without the **std::** prefix that would otherwise be needed.

```
1 #include <string>
2 #include <iostream>
3
4 using namespace std;
5
6 int main() {
7     string fName, mName, lName;
8     cerr << "First name: ";
9     getline(cin, fName);
10    cerr << "Middle name: ";
11    getline(cin, mName);
12    cerr << "Last name: ";
13    getline(cin, lName);
14    cout << "Hello, " << fName << " " << mName << " " << lName << "!"
15         << endl;
16    return 0;
17 }
```

5. (20 pts.) This question is about writing a **program that produces a specified number of bars of random lengths**, where a bar is a contiguous sequence of * characters on a line by itself. (Each line is terminated by single newline.) In more detail, provide **well-formatted source code of a complete C++ program** that:
1. Reads two non-negative integers (separated by one or more white-space characters) from *standard input* and stores them in suitably defined variables `numBars` and `maxLength` (in that order).

2. Writes `numBars` bars to *standard output*, with the length of each bar being a random integer between 0 and `maxLength` (both inclusive).

For example, if the two numbers in the input are 5 and 70, then the program should output 5 bars (so 5 lines) with each line containing a random number of `*`s between 0 and 70. It would look something like the following (depending on the random numbers produced):

```
*****
*****
*****
*****
*****
*****
```

Explain your answer briefly, especially to qualify for partial credit.

Ⓐ Random numbers are generated using the `rand` function from the `cstdlib` library with the generator seeded with the current time, provided by `time` from the `ctime` library. The program starts by reading the two numbers in the usual way. Then the core of the program is a pair of nested for loops. The outer loop iterates over bars while the inner one iterates over characters in each bar.

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4
5 using namespace std;
6
7 int main() {
8     int numBars, maxLength;
9     srand(time(NULL));
10    cin >> numBars >> maxLength;
11    for(int bar = 0; bar < numBars; bar++) {
12        int len = rand() % maxLength;
13        for(int col = 0; col < len; col++) cout << "*";
14        cout << endl;
15    }
16    return 0;
17 }
```