COS 220 Spring 2023 Quiz 2   *30 minutes; 30 pts.; 8 questions; 3 pgs.*   *2023-02-27 11:05 a.m.*

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.
   - Do not attach or remove any pages.

   Write your name in the space provided above.

   Do not write on this page below this point.

2. (2 pts.) Provide a single C++ statement that declares an array, named `a5`, of five unsigned integers and initializes it to contain the elements (in index order): 3, 1, 4, 1, 5.

   Ⓐ `unsigned int a5 = {3, 1, 4, 1, 5};`

3. (2 pts.) Provide a single C++ statement that prints, to *standard output*, the number of elements (items) in an array named `howMany`, whose elements are of type `float`.

   Ⓐ `std::cout << sizeof(howMany)/sizeof(float);`

4. (2 pts.) Provide a single C++ statement that declares an array, named `hislah`, containing four elements of type `char`, and initializes it to contain the elements (characters, in index order): `y`, `e`, and `s`.

   Ⓐ `char hislah[] = "yes";`

5. (2 pts.) Provide a single C++ statement that has the same effect as the one in Question 4, and that is *as different from that one as possible.*

   Ⓐ `char hislah[] = {’y’, ’e’, ’s’, 0};`

6. (6 pts.) Describe, as precisely as possible, the output produced by the following C++ program. (If the program will not compile, will crash, or otherwise not produce output, then explain why clearly.) *Explain the reason for your predicted output as precisely as possible; there is no credit for answers without explanations.*

```
#include <iostream>
int x[3];
int main() {
  int y[3], z[] = {1, 2, 3};
  for(int i=1; i < 3; i++) y[i] = z[i-1];
  for(int i=0; i < 3; i++) cout << 100*x[i] + 10*y[i] + z[i] << endl;
  return 0;
}
```

Ⓐ *The global array `x` is automatically initialized to contain three zeros as elements. The local array* y *is not initialized and contains unpredictable values as elements at first. The local array `z` is explicitly initialized to contain 1, 2, 3, in order. The first for loop assigns `y[1]` and `y[2]` using values of `z[0]` and `z[1]`, i.e., 2 and 3, respectively. The element `y[0]` has an unpredictable (undefined behavior) value. The second for loop computes and outputs an expression for each index* $0, 1, 2$. *This expression produces an unpredictable value for index* 0 *(due to `y[0]`) but gives predictable values 12 and 23 for indices* 1 *and* 2. *Thus the* ***output*** *of the program has an* unpredictable value on the first line and 12 and 23 on the second and third lines, *respectively.*

7. (5 pts.) Provide well-formatted C++ code that defines a function `rot3` that cyclically assigns each of its arguments to have the value of the next one. In more detail, the definition should ensure that the following program, which uses but does not define `rot3`, produces "1 4 3" as output.

```
#include <iostream>
// your rot3 definition goes here
int main() {
  int a = 3, b = 1, c = 4;
  rot3(a, b, c);
  std::cout << a << " " << b << " " << c << std::endl;
  return 0;
}
```

Ⓐ

```
void rot3(int & x, int & y, int & z) {
  int t = x;
  x = y;
  y = z;
  z = t;
}
```

8. (10 pts.) Provide **well-formatted source code of a complete C++ program** that:

   1. Prints a prompt "Integers: " to *standard error* (note, not standard output).

2. Reads five whitespace-separated integers from *standard input*. Note that the amount of separating whitespace is arbitrary and may include spaces, tabs, newlines, etc.
3. Stores what is read in a suitably named and defined *array* variable.
4. Writes six lines to *standard output*. The first five lines consists of the five integers read earlier, in input order and one per line. The last line consists of the sum of those integers.

Poorly formatted, messy, or otherwise hard to read code will result in very substantial loss of points. *Explain your answer briefly, especially to qualify for partial credit.*

Ⓐ *The code here defines and uses the* **`inputInts`** *array for the five integers read from standard input. The standard header* **`iostream`** *is needed for using the standard IO streams* **`cin`**, **`cout`**, *and* **`cerr`** *and the* **`using namespace`** *line allows them to be used without the* **`std::`** *prefix that would otherwise be needed. The array is iterated over twice using the newer form of for loops using a range variable* **`e`** *in both cases, which is declared as a reference type in the first one to ensure array elements are modified to store the value read from* **`cin`**. *The second loop uses* **`s`** *as an accumulator variable into which values from the array are summed.*

```
#include <iostream>

using namespace std;

int main() {
  int inputInts[5];
  cerr << "Integers: ";
  for(auto & e : inputInts) cin >> e;
  int s = 0;
  for(auto e : inputInts) {
    cout << e << endl;
    s += e;
  }
  cout << s << endl;
  return 0;
}
```