

EECE.3220: Data Structures

Spring 2017

Lecture 6: Key Questions

January 30, 2017

1. (Review) Describe the different methods of function argument passing supported in C++.

2. **Example:** Show the output of the following short program.

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

double f1(int v1, int v2);
void f2(int *ptr1, int *ptr2);
void f3(int &ref1, int &ref2);

int main() {
    int foo = 10;
    int bar = 57;
    double baz;

    baz = f1(foo, bar);
    cout << "After f1(), foo = " << foo << ", bar = "
         << bar << ", baz = " << baz << "\n";

    f2(&foo, &bar);
    cout << "After f2(), foo = " << foo << ", bar = " << bar << "\n";

    f3(foo, bar);
    cout << "After f3(), foo = " << foo << ", bar = " << bar << "\n";

    return 0;
}

double f1(int v1, int v2) {
    return (v1 + v2) / 2.0;
}

void f2(int *ptr1, int *ptr2) {
    while (*ptr1 > 5) {
        *ptr2 -= 3;
        (*ptr1)--;
    }
}

void f3(int &ref1, int &ref2) {
    if (ref1 == 5 && ref2 >= 45) {
        ref1++;
        ref2--;
    }
    else if (ref1 == 5) {
        ref1--;
        ref2++;
    }
    else {
        ref1 = ref2 - 10;
        ref2 = ref1 + 10;
    }
}
```


3. Explain the use of `setprecision`. Why is `fixed` necessary?

4. What is the output of the following program?

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

int main()
{
    double root2 = sqrt( 2.0 ); // calc square root of 2
    int places;                // precision, vary from 0-9
    cout << "Square root of 2 with precisions 0-9." << endl;

    cout << fixed; // use fixed point format (not sci. not)

    // set precision for each digit, then show square root
    for ( places = 0; places <= 9; places++ )
        cout << setprecision( places ) << root2 << endl;
    return 0;
}
```

5. Explain the stream manipulator `showpoint`.

6. Show the output of the program below if the input stream is:

1 2 3.4 5

```
// NOTE: The include and using statements are not shown
int main()
{
    double i, j, x, y;
    cin >> i >> j >> x >> y;
    cout << fixed << showpoint;
    cout << "First output " << endl;
    cout << i << ',' << j << ','
        << setprecision(3) << x << ',' << y << endl;
    return 0;
}
```

- 6