Tutorial / Exercise – Call Stack

Exercises:

1. Look at the following code snippets. What will be printed out in each case. Why?

   a.

```
void foo(int a, int b)
{
    a = 10;
    b = 20;
}

int main()
{
    int a = 5;
    int b = 2;
    foo(a, b);

    std::cout << a << std::endl << b;
}
```

   b.

```
int main()
{
    int my_number = 5;
    {
        int my_number = 7;
        {
            cout << my_number << endl;
            int my_number = 3;
            {
                int my_number = 15;
                cout << my_number << endl;
            }
        }
        cout << my_number << endl;
    }
}
```

2. Look at the following functions. In each case, how many bytes of data does each store on the stack from local variables and arguments.

   a.

```
void PrintNumbers(int i, int j)
{
    std::cout << i << j << std::endl;
}
```

   b.

```
float AverageArray()
{
    int arr[100];
    for (int i = 0; i < 100; ++i)
    {
        arr[i] = rand();
    }

    float average = 0;

    for (int i = 0; i < 100; ++i)
    {
        average += i;
    }
    average /= 100.0f;

    return average;
}
```

3. The following code will crash. Why?

```
void bar()
{
    double my_doubles[1024][1024];
}

void foo()
{
    int my_floats[1000][100];
    bar();
}


int main()
{
    bool my_bools[100][10];
    foo();
}
```

4. Write a program that causes a stack overflow.