Lauren Lyne

Hw02 questions

1.What is the difference between int and const int?

> An int is an integer that’s value can be changed by the programmer. A const int is an int whose value is set and then remains the same and may not be changed by the programmer.

2. What is the difference between a function declaration and a function definition?

> A function declaration is used to tell the compiler the name of variables and what type of variables they are. On the other hand, a function definition makes the program allocate memory for that variable that you are declaring.

3.

What is wrong with the following C++ code? How would you fix it?

```
   int sum;
   for(int i=0; i<1000; ++i) {

sum += i; }
   printf("Sum of 0 to 999 is %d\n", sum);
```

> The int sum was not initialized to a value. Unlike java, in C++, you need to initialize the variable because the compiler will not automatically set it to zero. In order to solve this error, the programmer should declare the sum to be 0.

4.
What is wrong with the following code and how would you fix it?

```
   int n = 1;
   if(n = 0) {

      print("n is zero\n");
   }
```

In C++, there is no such thing as just print. The compiler will not recognize that command. To change it so that the compiler can recognize it, you would change print to printf.

5.

What is wrong with the following code and how would you fix it?

```
int* ptr = nullptr;
scanf("%d", ptr);
printf("You entered %d", *ptr);
```

When using scanf, inside the parentheses there should be a “&” in front of ptr. Without that, you are not returning the pointer to the variable ptr. To change this you would simply make it &ptr inside the parentheses.

6.

What is the difference between the * and & operators?

The & operator returns a pointer to the variable while the * does the opposite. The * is dereferencing the pointer and returns the memory allocation that it points to.

7.

Describe what each of the following declare:

(a) int* a; A pointer to an integer named “a”

(b) const int b; An integer named “b” that is constant and the compiler won’t allow it to be changed

(c) const int* c; A pointer to a constant integer named “c”. You can change the pointer, but not the int variable it points to.

(d) int* const d; A constant pointer to a non constant int. You can modify the data points to it but not the pointer.

(e) const int* const e; A constant pointer to a constant int. Neither the data points or pointer can be modified.