**Exercise #2 – Pointers and References**

Handed out: 9/15/15

Due: 9/17/15

Questions

1. [5 pts] What is the difference between a definition and a declaration of a variable? Give two examples of declarations that are not definitions.

A definition is something that is given a value, it is more than a placeholder. A declaration is simply saying that something is going to be put there.

void myFunction(int x,int y); <----- In this instance, myFunction is being declared using a forward declaration. The function hasn't been fleshed out. Basically this is so the compiler knows that there is a function and what will be passed to it.

int num; <------ num is declared but not defined. There is a location in memory set aside for num, but no meaningful value has been put into that memory.

1. [4 pts] Are the following definitions valid? If not, why? If valid, what do they do?
   1. int ival = 1.01;

Valid, but it will truncate to just being 1 because 1.01 is a float but ival is an integer.

* 1. int &rval = ival;

Valid, assuming that ival has been defined and is visible to where rval is being defined. rval will be defined as an alias to ival.

* 1. int &rval1 = 1.01;

This is invalid because rval needs to be an alias for something defined, 1.01 is not a defined variable.

* 1. int &rval3;

This is invalid for the same reason, because aliases need

1. [3 pts] Are the following definitions valid? If not, why? If valid, what do they do?

int ival = 3, \*p = &ival, \*&r = p;

This is valid. ival is an integer, p is a pointer to ival and r is a reference to p. Dereferencing both p and r will give you 3.

1. [3 pts] What does the following code print?

int i, &ri = i; i = 5; ri = 10;

std::cout << i << “ “ << ri << std::endl;

The code prints 10 10 because the last line reassigns i and ri to 10.