

Name: _____

CS 161 Week 5 Worksheet: Functions, References, and Scope

Functions

1. What are pre-conditions and post-conditions? Pre-conditions and post-conditions become (slightly!) more important when the function in question has reference parameters. For each of the three following functions, write a brief description of their purpose, as well as their pre-conditions and post-conditions.

```
double calc_average(double sum, int count){  
    return sum/count;  
}
```

Description: return the average, given the sum and count

Pre-condition: sum is a double; count is a non-zero positive integer

Post-condition: the average, which is sum/count, is returned

```
void swap_chars(string& str) {  
    char first = str.at(0);  
    str.at(0) = str.at(1);  
    str.at(1) = first;  
}
```

Description: swap the first and second characters of the given string, str

Pre-condition: the string parameter is passed as a reference, and the string must have at least two characters

Post-condition: the first and second char of the string are swapped

```
void update_average(double &average, int &count, double new_val)  
{  
    double sum = average*count;  
    sum += new_val;  
    count += 1;  
    average = sum/count;  
}
```

Description: Update the average and count given a new_val as the value to be added

Pre-condition: average (double) and count (int) are passed by reference; count is a non-negative integer

Post-condition: the average and count are updated using the new value

2. What is a default argument, and where does it need to be?
A default argument is a parameter that is auto-assigned (to the value specified in the function definition) if the caller of the function doesn't provide a value for that argument. Any default arguments must come last in the parameter list.
3. Can you ever have multiple functions with the same name in C++? Explain.
Yes. Multiple functions with the same name can co-exist if they have different parameter lists (not just return types).

Variable References and Scope

1. What is your understanding of Pass by Value and Pass by Reference?

Pass by value: the value of the variable is passed to the function. If you change the value of the variable inside the function, the value is unchanged in the calling function.

Pass by reference: the memory address is passed to the function. If you change the value inside the function, the value in the calling function also gets updated.

2. With what you now know, how would you make the following code work?

```
void add_one(int& a) {
    a++;
}
int main() {
    int a = 5;
    add_one(a);
    cout << "5 plus 1 = " << a << endl;
    return 0;
}
```

Change the definition of add_one() to take parameter "a" as a reference (int& a).

3. Fill in the blank line with a function call to swap the values of a and b.

```
void swapnum(int& i, int& j) {
    int temp = i;
    i = j;
    j = temp;
}

int main() {
    int a = 10, b = 20;
    cout << "A is " << a << " and B is " << b << endl;
    swapnum(a, b);
    cout << "A is " << a << " and B is " << b << endl;
    return 0;
}
```

4. What will this program print?

```
int main() {
    int s = 17;
    if (s < 3)
        int s = 10;
    else
        int s = 3;
    cout << s << endl;
    return 0;
}
```

17 (discuss why)