**CS-1 Final Exam Study Guide**

**Exam 3 Info:**

PROJECT 1-2

C-Strings and Strings:

* C-String example:
  + Const int MAX = 25;
  + Int main()
  + {
  + Char name[MAX] = “ “;
  + Cout << “ Enter your full name: “ << endl;
  + Cin.getline(name, 25);
  + Cout << name << endl;
  + Return 0;
  + }
* String example:
  + #include <iostream>
  + #include <string>
  + Int main()
  + {
  + String name;
  + Cout << “Enter your full name << endl;
  + Getline (cin, name);
  + Cout << name << endl;
  + Return 0;

Arrays:

* + Contiguous Memory: Memory allocated as a single chunk
  + Index Values: array[**i]** = 10;
    - I is the index
  + Possible types of arrays: one, two, and three dimensional
  + Garbage Values: values in the array when they are not defined
  + Passing to functions:
    - Pass by reference: a way to pass the same value in different scopes. Arrays are always pass by reference.
    - Also pass in (usually) the size of the array
    - Ex: void function(int a[], int s)

Random Numbers:

* Srand(): from <cstdlib>, used to initialize a PRNG. It accepts a seed.
* Rand(): used to generate a series of random numbers
* Constraining numbers to a specific range:
  + Upper bound – lower bound + 1 = range
  + Rand()%range = randomNum;
  + randomNum +=1;

Searching:

* Linear search: going down the list of an array one by one.
  + Less efficient.
  + Example:
* Binary search: a method of searching by continuously halving and searching in the half.
  + More efficient
  + Example:

Sorting:

* Insertion Sort
  + Using comparison to sort a list by comparing the values to other parts of the list
  + You start adding values instead of having a full list
  + Worst case performance is O(n^2)
* Bubble Sort
  + Compare pairs of values and swap them if they are out of order (like a bubble rising to the top of water)
  + N-1 total repetitions
  + Total comparisons = (n-1)(n-1) which is also O(n^2) (its going to cost n^2 comparisons)

2D Arrays:

* Type arrayName[x][y];

**Exam 2 Info:**

PROJECT 3-

Repetition control structures (loops):

* While Loop
  + 3 parts:
    - initializing the counter
    - condition
    - update statement
  + Example:
    - While (condition)
    - { …..
    - if (condition 2)
    - {
  + }
  + (OR ANY VALID CODE)
  + }
  + **CHECK A8 and ICE-11**
  + Example 2:
    - int product = 3;
    - while(product > 0){
    - cout<<product<<endl;
    - product --;
    - }
* Do while loop:
  + How different from while: always executes at least once
  + check the loop continuation condition at the END of the loop
  + slight wrinkle in syntax as well
    - do{
    - stmt;
    - stmt;
    - stmt;
    - while(condition);
* For loops:
  + How different from while: 3 parts all in the header (more compact)
  + slight wrinkle in syntax as well
    - if(initialize; continuationCondition; update){
    - stmts;
    - }
* Nested loops:
  + 2D arrays for example:
  + For inside a for
* Sentinel Control (**SEE A10)**

File I/O:

* **SEE A11**
* FIRST, have to create the necessary “file streams” (outFile, and inFile)
* #include<fstream>
* New variable types to use:
  + ifstream – for input files
  + ofstream – for output files

Functions:

* **SEE A12 and Project 5 and Project 6 and ICE L-15**
* 3 parts to any function:
  + return type -- what TYPE of value (int, char, bool, double, etc.) does the function generate and send back
    - a NOTE about returning values
    - the return statement has to MATCH this
  + function name -- these are identifiers (same rules as variable names)
  + parameters -- the values that get sent to the method to facilitate processing
    - main has none of these
* Some functions are already defined for us:
  + - Pow function
    - Floor function
* User defined functions:
  + Declaring a function (declaration)the prototype (just the function header)
  + Defining a function (definition)the implementation (the code the function should execute)Invoking a function
  + Calling a function (making the function execute from your program)
  + 4 Types of User-defined Functions
    - no value returned, no value put in
    - no value returned, accepts value(s)
    - returns value, no value put in
    - returns value, accepts value(s)
    - **IF NOTHING IS RETURNED, IT IS A VOID FUNCTION**

Pass by value and Pass by reference:

* Pass by value: passing through a function within the scope
* Pass by reference: ability to change variables within different scopes using an &
* **SEE A-13**
* **SEE Exam 2 tracing problems and ICE L-16**

**Exam 1 info:**

* Look at A-1, A-2, L3, A-3, Day 1 and 2 Activity Sheet, A4a Unit Conversion, Project 1, A4, A5, L8, A6, A7, Project 2,