General Knowledge of

**Functions:**

* T/F If a function contains a statement that changes a value parameter, only the copy of the argument is changed, not the original.
* How do you declare a function?
* Where does the prototype go?
* What is the scope of a local identifier?
* T/F The scope of a value parameter is identical to the scope of a local variable declared in the outermost block of the function body.
* T/F In C++, corresponding arguments from a calling function and parameters from a called function must have the same name.
* When would you use a void function?
* How do you return multiple values from a function?
* T/F When passing by value data flow is one-way – into the function
* What type of arguments can be sent from the calling function when passing by reference?
* What type of arguments can be sent from the calling function when passing by value?
* T/F All functions should have a return statement.
* If a module is supposed to compute and return the average of five numbers, which is more appropriate a value returning function or a void function?
* T/F Functions must have a return data type
* What should the return data type for main be?
* What is the advantage of using functions?
* Declare a function that is passed two integers representing a sum and a count and returns an average.
* Declare a function that returns the sum & avg of an integer array.
* T/F You can only have 1 return statement in a function
* What is an advantage of passing by value?
* What is a disadvantage of passing by value?
* When we pass by reference what is passed to the calling function.
* What is a disadvantage of passing my reference?
* How do you pass by reference?

Header Files

* T/F You can put executable code in a header file.
* T/F It doesn’t matter what order the directives and declarations appear in a header file.
* What order should the directives appear and why?
* What are these lines in a header file for?

#ifndef MYHEADER\_H\_

#define MYHEADER\_H\_

#endif

* Does the extension for your header file matter?
* How do you include a user-defined header file?

Arrays

* What is an array?
* Is this a valid statement: firstArray = secondArray;
* Is this a valid statement: if(firstArray == secondArray)
* T/F an array is a composite data type.
* Why should we use constants for array size?
* How do we initialize an array?
* T/F An individual component of an array cannot be passed as an argument to a function. The entire array must be sent.
* How many different data types can you have in one array?
* T/F Given the declaration

***int someAr[20];***

***int someAr2[20];***

***cout << someAr[3];*** outputs the 3rd element in the array.

***cin >> someAr[20];*** will produce a compiler error

***someAr = someAr2;*** will transfer all values from someAr2 to

someAr

* What is the advantage of passing by reference?
* Which of the following is true about an array?

a) Arrays are always passed by reference.

b) The name of an array is the address in memory of the first element.

c) Array subscripts always begin at 0.

* How would you declare an array of 20 c-strings that can hold up to 11 characters?
* How would you compare an array of int.
* What will this statement do: int item[5]={2,12,1};
* What will this statement do: int item[5]={0};
* What will this statement do: int item[5]={2,12,1,2,9,5};
* T/F The compiler will give you an out of bounds error when using arrays if your index is too big or too small.
* What is stored in the array variable? (e.g. myArray)
* What is the base address?
* Know how to use loops with arrays
* T/F Arrays can be returned as a return value in a function.
* T/F Arrays must be passed by reference using the &.
* T/F When you pass an array you don’t have to include the size in the parameter list for the first dimension.
* How should you pass an array when you don’t intend to modify it in the function being called.
* T/F C-Strings are special arrays.
* T/F char name[16]= “Pete”; ⬄ char name[]= “Pete”;
* How would you declare parallel arrays that could contain a movie title, genre, and running time (in minutes)? Assume a const AR\_SIZE
* How would your read these values in from a file (assume a list with \n after each entry) – use the fstream variable inFile?
* How would you declare a multi-dimensional array that would hold 10 scores for 5 people?
* How would you read these values into the array from a file (assume the inFile variable is already assigned – assume 10 scores per row)?
* how to output multi-dimensional arrays and how to initialize them.

Structs

* What is the advantage of using structs?
* What is a member?
* Are aggregate operations allowed on structs?
* Can you pass structs by value or reference?
* Can structs be a return type?
* Define a struct called DvdRec, that contains the title, genre, and running time.
* Declare an array 100 elements of that struct called movies.
* How would you output the title of the 10th element in your array?
* Be able to write a function that can read into an array of structs or output an array of structs.

Pointers

* What is a pointer?
* Given the following contents in memory. What would these statements output?

cout << intPtr;

cout << \*intPtr;

cout << &intPtr;

* Know how to access members of a struct using pointers.

Classes

* What is the difference between Unstructured and Structured programming?
* What is the difference between Structured and OOP?
* OOP – focus is on data and methods are used to access data
* What is information hiding?
* What is the difference between a mutator and an accessor?
* What is a constructor?
* What is a destructor?

Command Line

* How to compile programs on the command line.
* How to compile without calling the linker.
* How to compile on the command line and set the executable file to something other than the default executable file (a.out).
* You should know some simple bash commands, such as ls and cd.
  1. Write the definition of a void function that takes as input a decimal number and as output 3 times the value of the decimal number. Format your output to two decimal places.
  2. Write the definition of a void function that takes as input two parameters of type int, say sum and testScore. The function updates the value of sum by adding the value of testScore. The new value of sum is reflected in the calling environment.

1. What is the output of the following program?

#include<iostream>

using namespace std;

void find(int a, int& b, int& c);

int main()

{

int one, two, three;

one = 5;

two = 10;

three = 15;

find(one, two, three);

cout << one << ", " << two << ", " << three << endl;

find(two, one, three);

cout << one << ", " << two << ", " << three << endl;

find(three, two, one);

cout << one << ", " << two << ", " << three << endl;

find(two, three, one);

cout << one << ", " << two << ", " << three << endl;

return 0;

}

void find(int a, int& b, int& c)

{

int temp;

c = a + b;

temp = a;

a = b;

b = 2 \* temp;

}

1. Identify error(s), if any, in the following array declarations.
   1. intlist[10];
   2. const int size = 100;
   3. double list[SIZE];
   4. int numList[0..9];
   5. string names[20];
   6. scores[50] double;
2. What is the output of the following code?

int list[] ={6, 8, 2, 14, 13};

for(int i = 0; i < 4; i++)

list[i]= list[i]- list[i + 1];

for(int i = 0; i < 5; i++)

cout << i << " " << list[i]<< endl;

1. Suppose that you have the following function definition.

void sum(int x, int y, int& z)

{

z = x + y;

}

Consider the following declarations:

int list1[10], list2[10], list3[10];

int a, b, c;

Which of the following function calls is valid?

a. sum(a, b, c);

b. sum(list1[0], list2[0], a);

c. sum(list1, list2, c);

d. for(inti = 1; i <= 10; i++)

sum(list1[i], list2[i], list[3]);

1. Define a two-dimensional array named temp of three rows and four columns of type int such that the first row is initialized to 6, 8, 12, 9; the second row is initialized to 17, 5, 10, 6; and the third row is initialized to 14, 13, 16, 20.
2. Given the declaration:

char str1[15];

char str2[15]= "Good day";

mark the following statements as valid or invalid. If a statement is invalid, explain why.

a. str1 = str2;

b. if(str1 == str2)

cout << " Both strings are of the same length." << endl;

c. if(strlen(str1) >= strlen(str2))

str1 = str2;

d. if(strcmp(str1, str2) < 0)

cout << "str1 is less than str2." << endl;

1. Write a C++ program to accept five integer values from keyword. The five values will be stored in an array using a pointer. Then print the elements of the array on the screen.
2. Write a program that asks the user to enter integers as inputs to be stored in the variables 'a' and 'b' respectively. There are also two integer pointers named ptrA and ptrB. Assign the values of 'a' and 'b' to ptrA and ptrB respectively, and display them.
3. Consider the following statements:  
   struct nameType{

string first;

string last;

};

struct dateType{

int month;

int day;

int year;

};

struct personalInfoType{

nameType name;

int pID;

dateType dob;

};

personalInfoType person;  
personalInfoType classList[ 100] ;  
nameType student;  
Mark the following statements as valid or invalid. If a statement is invalid, explain why.  
a. person.name.first = "William";  
b. cout << person.name << endl;  
c. classList[ 1] = person;  
d. classList[20].pID = 000011100;  
e. person = classList[ 20] ;  
f. student = person.name;  
g. cin >> student;  
h. for (int j = 0; j < 100; j++)

classList[j].pID = 00000000;  
i. classList.dob.day = 1;  
j. student = name;

1. Write a set of functions that take in a vector of ints and calculates the average, max, and min.
2. Declare an empty char vector named vowels. Then insert the letters a,e,i,o,u vowels into the vector.
3. Write a function vectorAverage that returns the average of the elements of an integer vector. The prototype is written below.

double vectorAverage (vector<int> v);

1. Write a function insertElement that inserts a given integer x into an integer vector before a given position pos. The prototype is written below.

void insertElement (vector<int> &v, int pos, int x);

1. Write a function replaceVowels that takes a string as input and replaces every lower-case vowel (aeiou) with an one of the four symbols below:

\* & # @

The symbol used should be selected at random. A sample run is shown below.

cout << replaceVowels ("Frodo, look out for the orcs!!!");

Output: Fr&d\*, l#\*k \*@t f&r th# &rcs!!!

1. Write a class Car that keeps track of the name of the car, the fuel efficiency in miles per gallon (mpg), the number of miles driven, and the number of gallons left in the gas tank. The constructor should take the car name and the fuel efficiency as input. Write the class declaration first and then define the member functions for the class. You'll have to figure out what functions to add based on the sample usage below.

Car my\_car ( "Ferrari Testarossa", 20 ); //My Ferrari gets 20 miles to the gallon.

my\_car.add\_gas (10); //Add 10 gallons of gas to the tank.

my\_car.drive (100); //Drive 100 miles.

cout << my\_car.get\_name( )<< " has driven " << my\_car.get\_miles( )

<< " miles and has " << my\_car.get\_gas( )

<< " gallons left in the tank.";

Output: Ferrari Testarossa has driven 100 miles and has 5 gallons left in the tank.