Signed – can be negative. Unsigned – can only be positive. Short – 2 bytes. char – 1 byte. int – 4 byte x86 , 8 byte x64.

Const – cannot be changed after initalisation. % - modulus(leftover from division) typedef- new name for datatypes

Static var/function – allocated,but not initialized until first run of the function/var. Global Static – private to file declared.

[] for array subscript can take int. {} for values, 2D = extra {} use for for loops to internate arrays, one for loop for each dimension of array.

Character arrays

Strcmp – compare 2 string arrays strcmp(string1,string2) strlen – find string length strlen(string1) strcpy – copies the contents of string 2 into 1.

Strcpy(string1, string2) strcat – concaternating two string arrays strcat(string1,string2)

Strings

<string> advant = no fixed size and helper functions, disadvant = no fixed size makes it slower to concate and append. + can be used for concatenation. Find string length = .length() or .size() can access characters in a string like an array with [thing]. Erase(starting pos, amount of characters ahead). Str.erase(0, str.length()) for whole string. .insert(starting pos , string itself)

Conditional Statements

Switch statements have cases based what what the value of the parameter passed. At the end of each case, there is a break. And default is the same as else.if no break, each case will carry to another. Ternary operator is result = (condition) ? True : False;

Loops

While – can run no times. Do while – is executed at least once. Break – breaks out of a loop. Continue = go to the top.

Functions

Functions need to be decelerated before they can be used by other functions. And this is done top to bottom. Overloading functions are functions with the same name but do different things based on params.

Pointers and mem

Unsigner char – 8 bits 0-255 unsigned shot – 16 bits 0-65,535 unsigned long – 32 bits 0-4,294,967,295 unsigned long long – 64 bits 0-18,446,744,073,709,551,615

& - returns the memory address that a var is stored at. \* = if in front of var, makes it a pointer, \* and & makes a pointer equal to a variable. For ex

Int my\_number;

Int\* my\_pointer = &my\_number; \* also returns the value at the address it is pointing at, called dereferencing the pointer.

* = dereference and accessing a variable within a struct or class.

The stack is a set size of memory for our program to allocate and deallocate all of the variables used. The heap is a much larger location that allows us to make use of RAM to store allocated variables

File I/O

Include fstream, open a file with file.open(“directory”, ios\_base::out); >> write to, << read from. File.close(); Binary files = more compact, easier to modifty, not human readable, less portable

Classes

Public – any function or class can access, protected – only functions from this class or derived or friend can access private – only functions from this class and friends can access

Deep copy – where all data the class holds is copied, including reference and pointers. Shallow is a copy from one object to another.

Template declaration

Template<typename A>

Type class(A)

Static Linked Library – a coolection of precompiled code that gets added when the link step happens. A DDL is the same by is done during runtime. DDL = exe is small, update DLL without needing to rebuild the dll. But can cause probs if dll is missing or is the wrong version.

Vectors

To find magnitude of vector – Pythagoras, then squarerooting the left number. Magnitude is respresented by ||V|| = ….

Normalizing is scaling the vector by its magnitude. So divid each component of the vector by the magnitude.

Dot product returns a vector scalar from two vectors. Simply multiply two vector’s x’s and y’s togeth and add them together. Example

A . B = AxBx + AyBy to return the perpendicular, swap the x and y components of the vector, then negate one based on if you want the perpendicular you want, (along the x or y)

Cross product example/explanation here…

Row Major vs Columm Major

Displacement is the distance travelled in a direction. It is the difference between two position vectors. This is represented by a small triangle.∆

Velocity is the rate at which displacement/position is changing. Average velocity is done with the formula v = ∆s / ∆t which is chance in position / change in time so out velocity is how far we moved divided by how long it took.

Acceleration is the rate at which velocity is changing. Is is worked out with ∆v / ∆t. which is velocity over time.

Finalvelo = initvelo + accerlation x time = finding a new velocity using current velocity

Displacement = initvelo x time + ½ acceleration x tie squared. = find displacement using initial velocity,time and acceleration.

Finalvelo squared = initvelo squared + 2 times acceration x displacement = find velocity if you don’t know time.

Displacement = ½ of (finalVelo + initvelo) x time = find the displacement using final velocity, initial velocity and time.

Force = mass times acceleration.

UML Stuff here….