2d Arrays

How do you declare a 2d array in C++?

int table[2][3]
Creates a table of 2 rows and 3 columns

How do you access data stored in a 2d array in C++?

table[1][0]
Access the item in the second row first column. Remember array indexing starts at 0 in C/++.

How do you write a function that takes a 2d array as a parameter?

void print_table(int table[][10]);
You must provide at least the column size when declaring a 2d array parameter.

Advanced Class Features

What is the purpose of a copy constructor?

A copy constructor allows an assignment operator = to be used to create a **copy** of an object rather than a **reference.**

Without copy constructor: Account acc2 = acc1; //Stores address of acc1 in acc2

With copy constructor: Account accs2 = acc1; //Creates a copy of acc1 data stored in acc2

Write a sample copy constructor declaration.

Account(const Account &copied_account);

Write two ways to call a copy constructor.

Account acc2(acc1);

Account acc2 = acc1;

Declare an example overloaded operator for << (stream operator).

Friend ostream & operator<<(ostream &OutputStream, const Account &acc);

What is a friend function? Why are friend functions generally not used?

A friend function is not a member of a class, but allows access to private data fields. They are generally not used because they “override” the principle of information hiding. Friend functions are, however, useful for tasks such as creating overloaded operators.

Arrays

How do you declare an array in C++?

int nums[] = { 5, 4, 3, 6, 8 };
char initials[5];

How do you pass an array into a function in C++?

function_name(array);

Note that you do not use brackets when passing an array into a function.

How is an array stored in memory?

An array is a continuous block of memory storing variables of the same data type. You can think of it like a bookshelf, or a row of cubby holes.

When would you want to use an array?

Arrays are useful when you have multiple items of the same data type that need to be stored. For example, a list of temperatures over a month would be a good use case for an array. You wouldn't want thirty variables!

As well, arrays are useful when the data needs to be ordered and needs to be accessed by index or searched in order.

Class Basics

How do you declare a class in C++?

```
class <name>
{
        //Body
};
```

Ex: class Account
{};

Describe what is meant by *private* and *public* fields in a class?

Private fields are only accessible to functions within that class. Public fields are accessible from functions outside of the class body.

How do you declare a constructor?

```
<classname>()
{
	//Body
}
```

Ex: Account()

How do you declare a destructor?

```
~<classname>()
{
	//Body
}
```

Ex: ~Account()

Write a basic setter for a float variable named balance.

```
void set_balance(float balance)
{
	this -> balance = balance;
}
```

Write a basic getter for a std::string variable named account_holder.

```
std::string get_account_holder()
{
	return account_holder;
}
```

How do you declare an instance of a class?

<classname> <variablename>;

or

<classname>* <variablename> = new <classname>();

Ex: Account bank_account;

Ex: Account* bank_account_ptr = new Account();

How do you call a constructor?

In the declaration as above:

<classname> <variablename>(<constructor args>);

or

<classname>* <variablename> = new <classname>(<constructor args>);

Ex: Account bank_account(initial_balance);

Ex: Account* bank_account_ptr = new Account(initial_balance);

Program Flow

Write a sample if/elseif/else statement in C++

```
if (num > 0)
{
	//Do something
}
else if (num < 0)
{
	//Do something
}
else
{
	//Neither above cases was true
	//Do something
}
```

Name 3 different types of loops in C++ and write 1 example

for loop
while loop
do while loop

```
for (int i = 0; i < 5; i++)
{
	//Do something
}
```

How do you declare a function with parameters in C++? Write and example.

int add(int num, int num_2);

void print(std::string item);

How do you call a function with parameters in C++? Write an example.

```
int a = 5;
int b = 6;
int c = add(a, b);
std::cout << c; //Prints 11
```

How do you indicate a reference parameter in a C++ function? Write an example.

void square(int& num); OR

void square(int & num);

void square(int &num);

How do you pass in a variable to a reference function? Write an example.

```
int a = 5;
square(a);
std::cout << a; //Prints 25
```

C++ Data Type Review

Name 3 ways to store a number in C++

int
float
double

How do you declare your own data type in C++?

typedef <data type> <your type name>;
typedef unsigned int index;

Write two ways to store the string “HI” in C++

char myString[] = { ‘H’, ‘I’, ‘\0’ }; or char myString[3];

```
#include <string>
…
string myString = “HI”;
```

Dynamic Objects

How do you dynamically allocate an object in C++?

Luckily, this is the same as any other data type! The only minor difference is that parenthesis can be used to call the constructor:

Test* test = new Test();

When you have an object pointer, how do you call member functions?

If we have a Test pointer called test, we use an arrow to call member functions

test -> getSomething();

Alternatively, a C style syntax with dot notation can be used, but this is very uncommon and ugly. Use the preferred arrow method. However, for the sake of completeness, this is what it looks like:

(*test).getSomething();

How does this contrast with a function call on a normal object instance (not allocated dynamically?)

With a normal object, dot notation is used:

test.getSomething();

What must always be done when dynamically allocated memory is no longer used?

The memory must always be freed! We don't want memory leaks in any software, let alone something in production software.

How is this done on an object pointer in C++?

delete test; // Frees the memory used by test back to the heap

Friend Functions

Declare an example overloaded operator for << (stream operator).

friend ostream & operator<<(ostream &OutputStream, const Account &acc);

For the last declaration, write a sample function body that outputs some information from the object. For example, write the overloaded operator to print the account holder's first and last name

```
friend ostream & operator<<(ostream &OutputStream, const Account &acc)
{
	OutputStream << acc.getFirstName() << " " << acc.getLastName();
}
```

How would you use the overloaded operator you declared above? Write an example

std::cout << myAcc;

//Prints Walter White

What is a friend function? Why are friend functions generally not used?

A friend function is not a member of a class, but allows access to private data fields. They are generally not used because they "override" the principle of information hiding. Friend functions are, however, useful for tasks such as creating overloaded operators.

Inheritance

Write a C++ class based on the following specifications

Use the following base class

```
class Lifter
{
        protected:

        int height;
        int weight;

        public:

        virtual void compete();
        void print_stats();
        int get_height();
        int get_weight();
}
```

Define a subclass called powerlifter. Add a field called total to the subclass. Make it so that all of the data members cannot be accessed publically. Make it so all of the public methods of the subclass can still be accessed publically. Using the principles of polymorphism, define a function that prints the height, weight and total for the lifter. Define the compete function so that it prints the lifter's total.

```
class Powerlifter : public Lifter
{
        private:

        int total;
```

```
    public:

    void print_stats()
    {
        std::cout<<"Height:  "<<Lifter::get_height()<<std::endl;
        std::cout<<"Weight: "<<Lifter::get_weight()<<std::endl;
        std::cout<<"Total: "<<total<<std::endl;
    }

    void compete()
    {
        std::cout<<"Total for this meet: "<<total<<std::endl;
    }
}
```

Basic Linked Lists

How can you visualize a linked list? Draw a simple linked list diagram.

Head → [data | next] → [data | next] → [data | next] → [data | null]

Write a linked list node structure.

```
struct node
{
    int data;
    node* next;
}
```

What does next point to in the last item of a linked list?

null

This is important, as it indicates to any methods that iterate over the linked list that the list is over. Otherwise, these methods would have no way of knowing where a list ends. Worse yet, if the last item points to a dangling pointer, this can cause serious errors.

More Advanced Object Oriented Features

Define Inheritance.

Deriving a class from an existing class. What is inherited depends on the type of inheritance specified:

From http://www.tutorialspoint.com/cplusplus/cpp_inheritance.htm

Public Inheritance: When deriving a class from a **public** base class, **public** members of the base class become **public** members of the derived class and **protected** members of the base class become **protected** members of the derived class. A base class's **private** members are never accessible directly from a derived class, but can be accessed through calls to the **public** and **protected** members of the base class.

Protected Inheritance: When deriving from a **protected** base class, **public** and **protected** members of the base class become **protected** members of the derived class.

Private Inheritance: When deriving from a **private** base class, **public** and **protected** members of the base class become **private** members of the derived class.

How do you declare a child class in C++?

class <derived> : <access> <base>

ex: class Square : public Shape

class Savings : protected Account

class Queue : private Linked_List

Define Polymorphism.

Polymorphism ("many forms") means having functions with the same names but different definitions in the base and sub classes.

ex: savings_account.print(); //Prints the balance

account.print(); //Prints the name and account number

Define late binding or runtime binding.

The program decides what functions to use for a class instance when the program is running rather than at compile time.

What keyword does late binding require in C++?

virtual

ex: virtual void print();

When would you use runtime binding?

You would use runtime binding when it doesn't make sense to define the function at compile time, as you might have a parent that cannot have a function definition that makes sense. You might also use it when you will be using a base class pointer to point to a subclass object.

ex: Shape shape;

shape.draw(); //How would we know what to draw? This doesn't make sense!

---

virtual void draw(); //Virtual prototype in base class Shape

void draw(); //In sub class Square

square.draw(); //Draws a square to the screen

More Class Basics

What is the difference between a public and private class member?

A private member can only be accessed by functions within the class itself, not by any outside functions such as main. If you try to use a private variable using dot notation, the compiler with throw an error message.

A public member can be accessed from inside and outside the class. Functions such as getters and setters are almost always public.

Why do we use getters and setters for object oriented programming rather than directly accessing member variables?

We want to maintain the abstraction or "information hiding" concept of object oriented programming. We often have variables in a class that the end user does not need to see. For example, if we are implementing a car simulator class, the user does not need to see variables that control the engine fuel pressure.

How do you dynamically allocate an object in C++?

MyClass* my_object = new MyClass();

Remember to free the memory when you are done with it using delete, such as:

delete my_object

What is a major difference between a class and a structure?

In a class, members are private by default.

In a structure, members are public by default (and are almost always used that way).

More Pointers

Start			End		
Memory Address	**Value**	**Name**	**Memory Address**	**Value**	**Name**
4	12	a	4	12	a
8	-51	b	8	4001	b
12	99	c	12	4000	c
16	24	d	16	4000	d
20	4000	e	20	50	e
24	20	f	24	12	f
28	13	g	28	4000	g
32	652	h	32	8	h

Operations

a = 12

g = e

f = a

*a = e

d = *a

h = &b

e = 50

b = *a + 1

Object Arrays

How do you declare an array of objects in C++?

<Class> <array name>[<num>]; or
<Class>* <ptr name> = new <Class>[<num>]

ex: Lifter members[45];
ex: Lifter* members = new Lifter[45];

How would you access a public method from an object in an array?

Ex: members[i].getMonthsRemaining();
Ex: members[i] -> getMonthsRemaining();

How would you access a public variable from an object in an array?

Ex: members[i].rate;
Ex: members[i] -> rate;

Name two differences between classes and structures

In a class, members are **private** by default where in a structure members are **public** by default.

Structures generally do not contain methods

Parallel Arrays

Describe what a parallel arrays are. Write a brief example

Parallel array store a record across multiple arrays using the same index in each array for one record. It is basically a table implementation with arrays.

Ex:

student_id[3] = 234;
name[3] = “Jesse Pinkman”;
major[3] = “Chemistry”;

Describe some advantages and disadvantages of parallel arrays

Advantages: Simple implementation, great for tabular data

Disadvantages: Unsuitable for large data sets (whole set has to reside in memory), easy to “break” with sort/update/delete type operations if not careful with indexes

Pointers

What does a pointer store?

A memory address.

How do you declare a pointer in C++?

<data type>* <name>;

Ex: int* int_address;

How do you retrieve the address of a variable in C++?

&<var>;

Ex: &int_var;

What does it mean to dereference a pointer?

To retrieve the data at the memory address stored by a pointer

How do you do so in C++?

*<pointer variable>

Ex: int int_var = 5;

int* int_ptr = &int_var;

std::cout<<*int_ptr; //Prints 5

Random Number Generation

What function must be called to "seed" the random number generator in C/++

srand(time(NULL));

Describe how to generate random numbers within a certain range.

rand() % max + start;

Recursion

What is a recursive function?

A function that calls itself.

Every recursive function needs what *cases*?

Every recursive function needs a *base case* or *stopping case* that ends the “loop” and “unwinds” the runtime stack back to the original function call. The stopping case allows the final answer to be returned to the original caller. Also needed is a *recursive case*, the case in which the “loop” continues by calling the function again.

What is iteration? Give an example of iteration in C++.

Iteration is just a simple loop. For loops, while loops, do while loops, etc. are all examples of iteration in C++. Other languages may include iterative constructs like for each loops to loop through arrays or lists.

Name a few common problems that are solved with recursion.

- Mathematical problems: factorials, Fibonacci numbers, greatest common divisors
- Sorting algorithms: merge sort, quick sort, radix sort, etc. can all be written recursively
- Data structures: Binary trees and variations

Given this example of a recursive factorial function, write in the recursive function call.

From: http://cis.stvincent.edu/html/tutorials/swd/index.html
Author: Br. David Carlson

```
/* Given:  n  A non-negative integer.
   Task:   To compute the factorial of n.
   Return: This factorial in the function name.
*/
long factorial(int n)
   {
   if ((n == 0) || (n == 1))   // stopping cases
      return 1;
   else                        // recursive case
      return n * factorial(n-1);
   }
```

Searching Algorithms

Briefly explain how linear search works

The algorithm moves through the array index by index and compares to the target value. If the value is not found, the algorithm generally returns something like -1.

Briefly explain how binary search works

The algorithm requires a sorted array. The middle of the array is found and compared to the target value. If it is greater or less, it discards the rest of the array and again finds the middle. This is continued until the item is found, or generally returns -1 if not found.

Describe the pros and cons of linear search

Pros: Very easy implementation, does not require any prior work on the array

Cons: Unsuitable for large data sets, slow [O(n)]

Describe the pros and cons of binary search

Pros: Suitable for larger data sets, fast [O(log n)]

Cons: Requires sorted data, more complex implementation

Sorting Algorithms

Name some pros and cons of bubble sort

Pros: Simple implementation, low code footprint

Cons: Unsuitable for large data sets, slow

Name some pros and cons of selection sort

Pros: Simple implementation

Cons: Always O(n^2) runtime (does not adapt to partially sorted data sets), unsuitable for large data sets.

Basic Stacks and Queues

Which data structure behaves as as first in first out (FIFO) data structure?

A queue

Which data structure behaves as a last in first out (LIFO) data structure?

A stack

What basic data structure do we usually use to implement a stack or a queue?

Since the size of our data frequently changes, we usually used a linked list data structure to implement a stack or a queue. This structure is easily resized and the insert operations can be modified to create our “push” and “pop” stack/queue operations.

Describe how we would “push” an item on to a stack using the above data structure. Describe the “pop” operation.

To push an item on to the stack, you would insert an item at the head of the linked list. To pop an item, you would remove the item at the head of the list. A queue would be similar, but you would insert items at the rear of the list and again remove from the head.

Text Files

How do you declare an input or output file stream object in C++?

ifstream myfile(“myfile.txt”);

ofstream myfile(“myfile.txt”);

How do you read data using the >> operator?

myfile >> data;

This is very similar to reading in from cin (getting data from the user on the command line). Cin is just another stream object!

When does the >> operator (an input stream) stop reading?

An input stream will read until whitespace is encountered using the >> operator.

How would you read until the end of the file in C++?

```
while(! myfile.eof())
{
        //loop body
}
```

Windows Programming

Describe the difference between pointers in standard C++ and Windows programming.

Standard C++ uses * to denote pointers, Windows programming uses ^

What parameter goes in the main function definition for a Windows Form application?

array<System::String ^> ^args

How do you access a form component using code?

From within the form code:

```
this->component.function();
this->component.property = value;
```

What has to prefix a string literal in Windows Forms programming?

L

L“Example String”

Name 3 data types in Windows Forms programming.

Int32
String^
Double

How do you initialize a Form from the main code file?

CharApplication::Run(gcnew Form1());