


CS150 Final Exam

Share

★ Leave the first rating

Terms in this set (138)

You got this!

Thanks

Which of these statements apply to C++?

- Produces native code that runs on the CPU
- Automatically catches errors like array out of bounds.
- Low-level language
- Compiles to bytecode
- More efficient than Java or Python
- Compiles to native code
- Interpreted by a virtual machine

Produces native code that runs on the CPU,  
More efficient than Java or Python,  
Compiles to native code

Assume a is 20 and b is 21; what prints?  
  

```
// 0123456789'123456789'123
string s = "The elephant in the room";
cout << s.substr(a, b) << endl;
```

- Runtime error
- "room"
- "room
- "r"

"room"

Look at the problem statement below. The\_\_\_\_\_ of the loop is that a period was encountered.  
  
How many characters are in a sentence? Count the characters in a string until a period is encountered. If the string contains any characters, then it will contain a period. Count the period as well.  
  

- None of these
- plan
- bounds
- goal

bounds

What manipulator can you use to ensure that his large floating-point number appear using regular decimal notation  
  

- hex
- decimal
- fixed
- setw
- setprecision
- scientific

fixed

CS150 Final Exam		Study	
<div><div>if (n % 2 == 1) cout &lt;&lt; "Odd" &lt;&lt; endl;</div><div><div>- multiple selection</div><div>- alternative action</div><div>- guarded action</div><div>- primed decision</div><div>- short-circuit decision</div><div>- none of these are correct</div></div></div>			
<div><div>Which line throws an exception because of range checking?</div><div>1. string s = "holey moley";</div><div>2. auto len = s.size();</div><div>3. auto a = s.front();</div><div>4. s.at(len) = a;</div><div>5. s[len] = 'c';</div></div> <div><div>- 3</div><div>- none of these</div><div>- 2</div><div>- 4</div><div>- 5</div></div>	4		
<div><div>Look at the problem statement below. The _____ of the loop is read a character and increment a counter.</div><div>How many characters are in a sentence? Count the characters in a string until a period is encountered. If the string contains any characters, then it will contain a period. Count the period as well.</div><div>- plan</div><div>- goal</div><div>- none of these</div><div>- bounds</div></div>	plan		
<div><div>"Use the insertion operator? What in Sam Hill is the insertion operator?" Do you know?</div></div>	<<		
<div><div>Examine the loop plan from your reader below. Line # 5:</div><div>1. Goal: count the characters in a sentence ending in a period</div><div>2. Given: the variable str is a string (may be empty)</div><div>3. Let counter = -1</div><div>4. If str has any characters Then</div><div>5. Let counter = 0</div></div> <div><div>- is the loop bounds</div><div>- None of these</div><div>- advances the loop</div><div>- is a goal precondition</div><div>- is a bounds precondition</div></div>	is a goal precondition		
<div><div>Examine the loop plan from your reader below. Line #6:</div><div>1. Goal: count the characters in a sentence ending in a period</div><div>2. Given: the variable str is a string (may be empty)</div><div>3. Let counter = -1</div><div>4. If str has any characters Then</div><div>5. Let counter = 0</div><div>6. Let current = first character in str</div></div>	is a bounds precondition		
<div><div>One-way, independent decision use:</div><div>- switch</div><div>- if</div><div>- if...if...else...else</div><div>- if...else...if...else</div><div>- if...else</div></div>	if		

CS150 Final Exam		Study	
<div><div>- update expression</div><div>- initialization statement</div><div>- condition expression</div><div>- None of these</div></div>			
<div><div>This compiles, runs and prints 5,4. What is the correct prototype?</div><div>int a = 4, b = 5;swap(a, b);cout &lt;&lt; a &lt;&lt; "," &lt;&lt; b &lt;&lt; endl;</div><div>- void swap(int&amp; a, int&amp; b);</div><div>- void swap(const int&amp; a, const int&amp; b);</div><div>- int swap(int a, int b);</div><div>- None of these</div></div>	<div>void swap(int&amp;a, int&amp; b);</div>		
<div><div>This compiles, runs and prints 12. What is the correct parameter declaration for x?</div><div>int x = 6;multiply(x, 2);cout &lt;&lt; x &lt;&lt; endl;</div><div>- int x</div><div>- const int&amp; x</div><div>- int&amp; x</div><div>- None of these</div></div>	<div>int&amp; x</div>		
<div><div>Look at the problem statement below. The _____ of the loop is to count the number of characters in a sentence. How many characters are in a sentence?</div><div>Count the characters in a string until a period is encountered. If the string contains any characters, then it will contain a period. Count the period as well.</div><div>- None of these</div><div>- bounds</div><div>- plan</div><div>- goal</div></div>	<div>goal</div>		
<div><div>In the classic for loop, loop control variables going from 0 to less-than n are said to employ:</div><div>- necessary bounds</div><div>- asymmetric bounds</div><div>- None of these</div><div>- intentional bounds</div><div>- symmetric bound</div></div>	<div>asymmetric bounds</div>		
Question #17 Mid-01	Thanks		
Question #18 Mid- 01	Thanks		
<div><div>Assume that the user enters: Steve 3.5 68</div><div>What value is stored in gpa?</div><div>- .5</div><div>- 3.5</div><div>- 68</div><div>- program crashes</div><div>- undefined</div></div>	<div>5</div>		
<div><div>Assume that name is a string object. Which of these expressions are legal?</div><div>- name == "sally"</div><div>- name = "sally" + name</div><div>- name += "fred"</div><div>- name = name + 777</div><div>- name &lt; "bob"</div><div>- "sally" += name</div><div>- name.equals("bob")</div><div>- Correct Answer name += 'X'</div></div>	<div>name == "sally"</div> <div>name = "sally" + name</div> <div>name += "fred"</div> <div>name &lt; "bob"</div> <div>name+= 'X'</div>		

CS150 Final Exam		Study	
<div>if (str == "quit") ... - True - False</div>			
<div>Assume c is a char variable. What value s stored in the variable a?  string s{"guten tag"}; auto len = s.size(); auto a = s.front(); s.at(len) = a; s[len] = c;</div>			'g'
<div>Assume a is 14 and b is 10; what prints?  string s{"feed the fish"}; cout &lt;&lt; s.substr(a,b) &lt;&lt; endl;  - Runtime error - "h" - "" - does not compile</div>			Runtime error
<div>What is stored in s after this code runs? string s{"xyzw"}; s.at(2) = 'Y';  - This is illegal because you cannot change the individual characters inside a string object. - xYzw - This is illegal because you cannot have a function on the left-hand side of an assignment. - xyYw - xyzw</div>			xyYw
<div>- Associates a name with a type - Read a value and store it in a variable - Copy a new value into an existing variable assign - Allocates space for a variable define - Provides a starting value when a variable is created initialize - A named storage area that holds a value variable</div>			declare input assign define initialize variable
<div>- Shorthand assignment - Post increment - Undefined behavior - Widening conversion - Pre decrement - Chained assignment - Narrowing conversion - Mixed-type expression</div>			y += z; x++; x = z++ - ++z; double a = y; --z; x = y = z = 10; z = 3.15; auto v = x*2.3;
<div>Which operator is used to see if all of a set of conditions is true? - logical not - conditional operator - equality - logical or - none of these - logical and</div>			logical and
Question #28 Mid- 01			thanks

CS150 Final Exam		Study	
<pre>int which; cin &gt;&gt; which; cout &lt;&lt; "The elephant in the room "; if (which % 2) cout &lt;&lt; "is white!" &lt;&lt; endl; else cout &lt;&lt; "is pink!" &lt;&lt; endl;</pre> <div><div>- None of these</div><div>- The elephant in the room is pink!</div><div>- The elephant in the room is white!</div><div>- The elephant in the room is white! The elephant in the room is pink!</div></div>			
<p>What happens here? (Carefully check each operator and semicolon.)</p> <pre>int y = 4; if (y &lt; 0); y = -y; cout &lt;&lt; y &lt;&lt; endl;</pre> <div><div>- Runtime error</div><div>- You Answered 4</div><div>- Syntax error</div><div>- Output is undefined.</div><div>- -4</div></div>	-4		
<p>How do you call the function shown here?</p> <pre>string square(int a) { return to_string(a * a); }</pre> <div><div>- All of these are legal.</div><div>- None of these are legal.</div><div>- int a = square(4);</div><div>- string a = square(42);</div><div>- string a; a.square(3);</div><div>- double a = square(4.0);</div></div>	string a = square(42);		
<p>What is printed when this runs?</p> <pre>int main() { int a = 3, b== ++a; cout &lt;&lt; "a-&gt;" &lt;&lt; a &lt;&lt; ", b-&gt;" &lt;&lt; b &lt;&lt; endl; }</pre> <div><div>- a-&gt;4, b-&gt;3</div><div>- a-&gt;4, b-&gt;4</div><div>- Anything, this is undefined behavior.</div><div>- This is a syntax error.</div></div>	a -> 4, b-> 4		
<p>What is printed when this runs?</p> <pre>int a = 3; int b = ++a - a++; cout &lt;&lt; "b-&gt; " &lt;&lt; b &lt;&lt; endl;</pre> <div><div>- 0</div><div>- 2</div><div>- 1</div><div>- Anything at all because this operation is undefined</div></div>	Anything at all because this operation is undefined		
A function that calculates and returns a value	fruitful function		

CS150 Final Exam		Study	...
A named block of code that carries out an action or calculates a value		function	
Another name for a function declaration		prototype	
Variables define along with the function to receive input		parameters	
Executing, running or invoking the function		calling	
A function that carries out an action instead of calculating a value		procedure	
Produces a value when the function is invoked		return statement	
Specifying the calculation fractions that occur when the function is used		defining	
Specifying the function name, type and parameter types.		declaring	
Value passed to the function when it is invoked		arguments	
(T/F) Implementation files may use the statement using namespace std;		True	
(T/F)An undefined error message is a linker error.		True	
(T/F) Implementation files must explicitly qualify each name from the standard library with std::		False	
(T/F) An undeclared error message is a linker error.		False	
In a library, the client or test program:  - None of these - consists of function definitions - You Answered consists of - instructions that produce the executable - consists of function calls - consists of declarations or prototypes		consists of function calls	
In a library the interface file: - consists of function definitions - consists of instructions that produce the executable - None of these - consists of declarations or - prototypes - consists of function calls		consists of declarations or prototypes	
- Append output to a file named z - Discard both output and errors rm - Write output to a new file named - Read the input from the file named z - Write errors to a new file named z - Send the output to the input of the program named z		pwd >> z rm x >/dev/null 2>&l z pwd > z cat < z cat x 2 > z date   z	

CS150 Final Exam		Study	...
(T/F) In a while loop, (condition) is followed by a semicolon.	False		
(T/F) Default arguments let you call a single function in several different ways.	True		
(T/F) Function overloading lets you call a single function in several different ways.	False		
<div>What prints? void fn(int, double, double&amp;) { cout &lt;&lt; "A" &lt;&lt; endl; } void fn(int, int, double&amp;) { cout &lt;&lt; "B" &lt;&lt; endl; } void fn(int, int, double) { cout &lt;&lt; "C" &lt;&lt; endl; } void fn(int, int, int) { cout &lt;&lt; "D" &lt;&lt; endl; } int main(){ fn(1, 2, 3, 4);} - Syntax error: no candidates - B - A - C - Syntax error: ambiguous - D</div>	Syntax error: no candidates		
<div>Different functions that have the same name, but take different arguments, are said to be: - overloaded - derived - overridden - covariant - default</div>	overloaded		
<div>Given the overloaded functions prototypes and the variable definition below, which of the function calls will fail to compile?  int f(int&amp;); int f(int); int f(int, int); int a = 7; - f(a); - None of these fail to compile - f(2.0); - f(3); - f('a','b')</div>	f(a);		
<div>Which line in the function "skeleton" below contains an error? #include "digits.h" // 1. int firstDigit(int n) // 2. { // 3. return ""; // 4. } // 5.</div>	#4		
<div>Which prototype(s) in the following header file are syntactically correct (legal)? #ifndef EXAMPLE_H #define EXAMPLE_H #include &lt;string&gt;  string f1(int a); int f2(double); void f3(std::string&amp; s, int n); double f4(); #endif</div>	f3 f4 f2		

CS150 Final Exam		Study	...
<ul style="list-style-type: none"><li>- reference parameters</li><li>- default parameters</li><li>- default arguments</li><li>- mandatory arguments</li><li>- overloaded arguments</li></ul>			
<p>Which of these prototypes is the best one to use in this circumstance?</p> <pre>int main(){ string str{"To be or not to be."}; cout &lt;&lt; "Most common letter is " &lt;&lt; mostCommon(str) &lt;&lt; endl;} - char mostCommon(const string&amp;); - char mostCommon(string); - char mostCommon(const string); - char mostCommon(string&amp;); - Any of these are fine. - None of these are correct</pre>	<pre>char mostCommon(const string&amp;);</pre>		
<p>If f() needs to change the argument passed here, the parameter must be declared as:</p> <pre>void f( . . . str); int main(){ string s = "hello"; f(s);} - const string - const string&amp; - string - string&amp; - It is not possible for f() to change the argument passed here.</pre>	<pre>string&amp;</pre>		
<p>Examine this code. Assume that there are no logic errors. Which is the best prototype?</p> <pre>int age; string name = read("Enter your name, age: ", age); - string read(string, int); - string read(const string&amp;, int&amp;) - None of these - string read(const string&amp;, int) - string read(const string, int&amp;)</pre>	<pre>string read(const string&amp;, int&amp;)</pre>		
<p>Given the overloaded functions prototypes and the variable definition below, which of the function calls will fail to compile?</p> <pre>int f(int&amp;); int f(int); int f(int, int); int a = 7; - None of these fail to compile - f(a); - f(2.0); - f(3) - f('a', 'b')</pre>	<pre>f(a);</pre>		
<p>(T/F) Counting the number of words in input by counting word transitions is an example of a state filter.</p>	<pre>True</pre>		
<p>(T/F) A state filter learns something about the stream by examining characters.</p>	<pre>True</pre>		
<p>(T/F) To test if an I/O operation succeeded you must explicitly call the stream's fail() member function.</p>	<pre>False</pre>		



CS150 Final Exam		Study	...
<p>What does this filter do?</p> <pre>char ch; while (cin.get(ch)) { if (isspace(ch) &amp;&amp; isspace(cin.peek())) continue; cout.put(ch);}</pre> <p>- Compresses spaces in a line and single-spaces lines of input</p> <p>- Compresses spaces to a single space only</p> <p>- None of these</p> <p>- Single spaces input lines only</p>	compresses spaces in a line and single-spaces lines of input		
<p>What does this filter do?</p> <pre>char ch; while (cin.get(ch)) { if (isalpha(ch)    isdigit(ch)    isspace(ch)) cout.put(ch);}</pre> <p>- None of these</p> <p>- Prints all characters from input</p> <p>- Prints only alphanumeric characters and spaces from input</p> <p>- Prints only non-alphanumeric characters from input</p>	Prints only alphanumeric characters and spaces from input		
<p>What does this filter do?</p> <pre>bool printing = false; char ch; while (cin.get(ch)) { if (printing) cout.put(ch); if (ch == '\n') printing = !printing; }</pre> <p>- Removes all of the newlines from a text file</p> <p>- Prints every other line of text, starting with the first</p> <p>- None of these</p> <p>- Prints every other line of text, starting with the second</p>	Prints every other line of text, starting with the second		
<p>What does this filter do?</p> <pre>char ch; while (cin.get(ch)){ if (isspace(ch)) continue; cout.put(ch);}</pre> <p>- Prints only spaces</p> <p>- Removes all spaces from input; prints a single line of output</p> <p>- Removes all spaces from input; prints each line separately</p> <p>- None of these</p>	Removes all spaces from input; prints a single line of output		
<p>What kind of error is this?</p> <pre>ex1.cpp:6:12: error: no viable conversion from 'int' to 'string' string a = 15; - None of these - Linker error (something is missing when linking) - Runtime error (throws exception when running) - Operating system signal or trap - Compiler error (something is missing when compiling) - Syntax error (mistake in grammar) - Type error (wrong initialization or assignment)</pre>	Type error (wrong initialization or assignment)		

CS150 Final Exam		Study	
'std::out_of_range'			
<ul style="list-style-type: none"><li>- Runtime error (throws exception when running)</li><li>- Linker error (something is missing when linking)</li><li>- Compiler error (something is missing when compiling)</li><li>- Type error (wrong initialization or assignment)</li><li>- Operating system signal or trap</li><li>- None of these</li><li>- Syntax error (mistake in grammar)</li></ul>			
(T/F) Calling a template function like to_string<int>(3.5) is known as implicit instantiation.		False	
(T/F)Without try and catch, the throw statement terminates the running program.		True	
(T/F) The line: ifstream in("x"); throws a runtime exception if a file x cannot be found.		False	
(T/F) Functions with generic parameters may use the keyword class or the keyword struct for their type parameters.		False	
Which of the following blocks is designed to catch any type of exception? <ul style="list-style-type: none"><li>- catch(){ }</li><li>- catch(*){ }</li><li>- catch(exception){ }</li><li>- catch(...){ }</li></ul>		catch(...){ }	
In a sequence of try/catch blocks, the last catch block of that sequence should be ____. <ul style="list-style-type: none"><li>- catch(int x){ }</li><li>- catch(str){ }</li><li>- catch(...){ }</li><li>- catch(exception){ }</li></ul>		catch(...){ }	
The class ____ is designed to deal with illegal arguments used in a function call. <ul style="list-style-type: none"><li>- illegal_argument</li><li>- bad_argument</li><li>- invalid_call</li><li>- invalid_argument</li></ul>		invalid_argument	
What prints? <pre>string s("hello"); try { if (s.size() &gt; 5) throw s.size(); if (isupper(s.back())) throw s.back(); if (s == "hello") throw string("hello"); s.at(s.size()) = 'x'; cout &lt;&lt; "one\n";} catch (const string&amp; e) { cout &lt;&lt; "two\n"; } catch (exception&amp; e) { cout &lt;&lt; "three\n"; } catch (...) { cout &lt;&lt; "four\n"; } - Undefined - one - three - four - two</pre>		two	

CS150 Final Exam		Study	...
<pre>try { if (s.size() &gt; 2) throw s.size(); if (islower(s.back())) throw s.back(); if (s == "hello") throw string("hello"); s.at(s.size()) = 'x'; cout &lt;&lt; "one\n";} catch (const int&amp; e) { cout &lt;&lt; "two\n"; } catch (const string&amp; e) { cout &lt;&lt; "three\n"; } catch (exception&amp; e) { cout &lt;&lt; "four\n"; } catch (...) { cout &lt;&lt; "five\n"; } - Undefined - two - four - one - five - three</pre>			
(T/F) The return value of the getline() function is a string object.	False		
(T/F) A token is a "chunk of meaningful data".	True		
(T/F) If an input stream's file is missing when you try to open it, its fail()member function returns true.	True		
(T/F) To read a line of text, you include the header file <string>.	True		
<p>This loop:</p> <pre>char c; while (in.get(c)){ cout &lt;&lt; c &lt;&lt; endl;}</pre> <ul style="list-style-type: none"><li>- illustrates raw character I/O</li><li>- illustrates token-based stream processing</li><li>- has a syntax error</li><li>- illustrates line-based stream processing</li><li>- is an endless loop</li></ul>	illustrates raw character I/O		
<p>Which line opens the file input.txt for reading?</p> <ul style="list-style-type: none"><li>- fstream in("input.txt");</li><li>- ifstream in("input.txt");</li><li>- iostream in = "input.txt";</li><li>- istream in("input.txt");</li><li>- istream in = "input.txt";</li></ul>	ifstream in("input.txt");		
<p>The file temp.txt contains "Orange Coast College". What prints?</p> <pre>ifstream in("temp.txt"); char c; while (in.get(c)){ if (isupper(c)) cout &lt;&lt; toupper(c);}</pre> <ul style="list-style-type: none"><li>- oRANGE cOAST cOLLEGE</li><li>- ORANGE COAST COLLEGE</li><li>- occ</li><li>- range oast ollege</li><li>- OCC</li></ul>	OCC		
<p>After opening the input stream in, which of these cannot be used to see if the file was successfully opened?</p> <ul style="list-style-type: none"><li>- if (in.bad()) {/ <b>couldn't open</b> /}</li><li>- if (in.good()) {/ <b>opened ok</b> /}</li><li>- if (in) { / <b>opened ok</b> / }</li><li>- if (in.fail()) {/ <b>couldn't open</b> /}</li><li>-if (in.opened()) {/ <b>opened ok</b> /}</li></ul>	if (in.opened()) {/ <b>opened ok</b> /}		

CS150 Final Exam		Study	
<pre>{ cout &lt;&lt; item &lt;&lt; endl;} - print("hello"); - print(2 + 2); - None of these - print(3 + 2.2); - print(string("goodbye"));</pre>			
(T/F) You may create a structure variable as part of a structure definition.	True		
(T/F) This is the correct syntax for a C++ scoped enumeration. enum class WEEKEND {SUNDAY, SATURDAY=6};	True		
(T/F) When passing a structure variable to a function, use non-const reference if the intent is to modify the actual argument.	True		
(T/F) Structures data members may each have a different type.	True		
Given the following structure and variable definitions, which data members are default initialized? struct Employee { long empID; std::string lastName; double salary; int age;}; Employee bob{777, "Zimmerman"}; - None of these - You Answered empID - Correct Answer age - lastName - salary	salary		
Examine the following code (which is legal). Which statement is illegal (given only this code)? struct Money { int dollars{0}, cents{0}; } m1, m2; - if (m1.cents != m2.dollars) . . . - m2.cents++; - m1 = m2; - cout << m1 << endl;	cout << m1 << endl;		
Given the following structure and variable definitions, which data members are uninitialized? struct Employee { long empID{0}; std::string lastName; double salary{0}; int age = 0;}; Employee bob; - lastName - empID - None of them (does not compile) - None of them (compiles) - salary - age	None of them (compiles)		
Given the following structure and variable definitions which statements are legal? struct Money { int dollars{0}; int cents{1}; }; Money payment; - payment.cents = 5; - None of them - payment{1} = 5; - Money{1} = Money{0}; - cout << Money.dollars; - cout << payment.dollars;	payment.cents = 5; cout << payment.dollars;		
Question #59 Mid-02	...		

CS150 Final Exam		Study	...
<pre>for (auto e : s){ if (toupper(e)) out.put('x');}</pre> <ul style="list-style-type: none"><li>- sentinel loop</li><li>- limit loop</li><li>- counter-controlled loop</li><li>- primed loop</li><li>- data loop</li><li>- iterator or range loop</li><li>- loop-and-a-half</li><li>- inline test</li></ul>			
<p>Which of the following loop patterns are used here?</p> <pre>int upper = 0;char ch;while (in.get(ch)) { if (ch &gt;= 'A' &amp;&amp; ch &lt;= 'Z') upper++;}</pre> <ul style="list-style-type: none"><li>- primed loop</li><li>- limit loop</li><li>- iterator or range loop</li><li>- inline read and test</li><li>- loop-and-a-half</li><li>- counter-controlled loop</li><li>- sentinel loop</li><li>- data loop</li></ul>	<p>inline read and test</p> <p>data loop</p>		
<p>What is stored in data after this runs?</p> <pre>vector&lt;int&gt; data{1, 2, 3};data.back();</pre> <ul style="list-style-type: none"><li>- []</li><li>- [2, 3]</li><li>- [1, 2, 3, 0]</li><li>- None of these</li><li>- [1, 2]</li><li>- [1, 2, 3]</li></ul>	<p>[1, 2, 3]</p>		
<p>Below is a declaration for a partially-filled array. What is the correct prototype for a function insert() that inserts a new element at position pos in the array, shifts the remaining elements right, and returns true if successful?</p> <pre>const size_t MAX = 100;double nums[MAX];size_t size = 0;</pre> <ul style="list-style-type: none"><li>- bool insert(double a[], size_t size, size_t MAX, double e, size_t pos);</li><li>- bool insert(double a[], size_t MAX, double e, size_t pos);</li><li>- None of these</li><li>- bool insert(double a[], size_t&amp; size, size_t MAX, double e, size_t pos);</li><li>- bool insert(double a[], size_t&amp; size, double e, size_t pos);</li></ul>	<pre>bool insert(double a[], size_t&amp; size, size_t MAX, double e, size_t pos);</pre>		
<p>Which statement displays the element appearing in the second row and the third column?</p> <ul style="list-style-type: none"><li>- cout &lt;&lt; a[3][2];</li><li>- cout &lt;&lt; a[2][1];</li><li>- None of these</li><li>- cout &lt;&lt; a[1][2];</li><li>- cout &lt;&lt; a[2][3];</li></ul>	<pre>cout &lt;&lt; a[1][2];</pre>		
<p>Which lines create the C-string "hello"?</p> <pre>1. char s[10] = "hello"; 2. char s[10] = {'h','e','l','l','o'}; 3. char s[] = {'h','e','l','l','o','O'}; 4. char s[5] = "hello"; 5. char s[] = "hello";</pre> <ul style="list-style-type: none"><li>- 1, 3</li><li>- 1, 5</li><li>- 1, 2, 5</li><li>- All of them</li><li>- 1, 2, 3, 5</li></ul>	<p>1, 2, 5</p>		

<div>condition?</div> <div>const size_t MAX = 100;</div> <div>double nums[MAX];</div> <div>size_t size = 0;</div> <div>- while (size &lt; MAX) . . .</div> <div>- while (MAX &lt; size) . . .</div> <div>- for (size = 0; size &lt; MAX; size++) . . .</div> <div>- while (size &lt;= MAX) . . .</div> <div>- None of these</div>	
<div>Which area of memory is your program code stored in?</div> <div>- Text</div> <div>- Uninitialized Data</div> <div>- Initialized Data</div> <div>- Stack</div> <div>- Heap</div>	<div>Text</div>
<div>void f(const vector&lt;int&gt;&amp; v){ v.at(0) = 42;}</div> <div>int main(){ vector&lt;int&gt; x{1, 2, 3};</div> <div>f(x);</div> <div>cout &lt;&lt; x.at(0) &lt;&lt; endl;}</div> <div>- 42</div> <div>- Nothing; compile-time error.</div> <div>- Nothing; run-time error.</div> <div>- 1</div> <div>- Nothing; linker error</div>	<div>Nothing; compile-time error.</div>
<div>What is printed here? (Assume all includes have been added.</div> <div>Assume 4-bytes per int, 8 bytes per pointer.)</div> <div>size_t len(const int a[])</div> <div>{ return sizeof(a) / sizeof(a[0]);}</div> <div>int main(){ int a[] = {2, 4, 6, 8};</div> <div>cout &lt;&lt; len(a) &lt;&lt; endl;}</div> <div>- 2</div> <div>- Does not compile</div> <div>- 4</div> <div>- 1</div>	<div>2</div>
<div>The variable p:</div> <div>void f(){ int *p = new int;}</div> <div>- is uninitialized</div> <div>- stores the value 0</div> <div>- None of these</div> <div>- stores a memory address</div>	<div>stores a memory address</div>
<div>Which returns the last pixel on the first row of this image?</div> <div>Pixel *p; // address of pixel data</div> <div>int w, h; // width and height of image</div> <div>- p + w - 1</div> <div>- None of these are correct</div> <div>- *p + w - 1</div> <div>- *(p + w - 1)</div> <div>- *(p + w) - 1</div>	<div>*(p + w - 1)</div>

CS150 Final Exam		Study	...
<pre>const T* p, size_t n) { out &lt;&lt; '['; if (n) { out &lt;&lt; p[0]; for (size_t i = 1; i &lt; n; i++) out &lt;&lt; ", " &lt;&lt; p[i]; } out &lt;&lt; "]; return out;}. . int a[] = {1,2,3,4,5,1}; mystery(cout, a, sizeof(a) / sizeof(a[0])) &lt;&lt; endl;</pre> <div><div>- [1, 2, 3, 4]</div><div>- [1, 2, 3]</div><div>- None of these or undefined output.</div><div>- [1, 2, 3, 4, 5, 1]</div><div>- [1, 2, 3, 4, 5]</div></div>			
<p>Below is a declaration for a partially-filled array. What is the correct prototype for a function delete() that deletes the element at position pos in the array, shifts the remaining elements left, and returns true if successful?</p> <pre>const size_t MAX = 100; double nums[MAX]; size_t size = 0;</pre> <div><div>- None of these</div><div>- bool delete(const double a[], size_t&amp; size, size_t pos);</div><div>- bool delete(double a[], size_t size, size_t pos);</div><div>- bool delete(double a[], size_t&amp; size, size_t pos);</div><div>- bool delete(double a[], size_t MAX, size_t&amp; pos);</div></div>	<pre>bool delete(double a[], size_t&amp; size, size_t pos);</pre>		
<p>Which expression returns the number of countries?</p> <pre>string countries[] = {"Andorra", "Albania", . . . }; - sizeof(countries) - None of these - sizeof(countries) * sizeof(countries[0]) - sizeof(countries) / sizeof(countries[0]) - len(countries)</pre>	<pre>sizeof(countries) / sizeof(countries[0])</pre>		
<p>This code:</p> <pre>int * f() { int a[] = {1, 2, 3}; return &amp;a[1];} - has a dangling pointer - has a double delete - has a syntax error - None of these - has a memory leak</pre>	<p>has a dangling pointer</p>		
<p>Which of these lines correctly prints 3?</p> <pre>struct S { int a = 3; double b = 2.5;}; S obj, *p = &amp;obj; - cout &lt;&lt; (*p).a &lt;&lt; endl; - cout &lt;&lt; *p.a &lt;&lt; endl; - cout &lt;&lt; p.a &lt;&lt; endl; - cout &lt;&lt; *(p.a) &lt;&lt; endl; - cout &lt;&lt; *(p).a &lt;&lt; endl;</pre>	<pre>cout &lt;&lt; (*p).a &lt;&lt; endl;</pre>		
<p>Which line compiles, but crashes when run?</p> <pre>int main(){ vector&lt;int&gt; v{1, 2, 3}; auto size = v.size(); cout &lt;&lt; v.back() &lt;&lt; endl; // 1. cout &lt;&lt; v.front() &lt;&lt; endl; // 2. cout &lt;&lt; v.at(0) &lt;&lt; endl; // 3. cout &lt;&lt; v.at(size) &lt;&lt; endl; // 4. cout &lt;&lt; v.pop_back() &lt;&lt; endl; // 5.}</pre>	<p>4</p>		

cout << sizeof(a) << endl;  
- Illegal declaration. Does not compile.  
- 12  
- 8  
- 16  
- 4

What is the equivalent array notation?  
int dates[10];  
cout << (\*dates) + 2 << endl;  
- dates[0] + 2  
- dates[2]  
- dates[0] + 4  
- dates[2] + 2  
- &dates[2]

dates[0] + 2

What happens here?  
char s1[] = "CS150",  
s2[10];strcpy(s1, s2);s2[0] = 'X';cout << s1 << endl;  
Crashes when run.  
"XS150"  
Correct Answer Undefined behavior  
You Answered "CS150"  
Does not compile

...

Assume that ppi correctly points to pi. Which line prints the address of ppi?  
int main(){ double pi = 3.14159;  
double \*ppi;  
// code goes here // code goes here}  
- cout << &pi;  
- cout << \*ppi;  
- cout << &ppi;  
- None of these  
- cout << ppi;

cout << &ppi;

Where are the characters "Hello" stored in memory?  
char s1[1024] = "Hello";  
void f(){ const char \*s2 = "Goodbye";  
char s3[] = "CS 150",}  
- stack  
- heap  
- None of these  
- static storage area (read-only)  
- static-storage area (read/write)

static-storage area (read/write)

Below is a cumulative algorithm using an array and a range-based loop. What is printed? (Assume this is inside main() with all includes, etc.)  
int a[] = {2, 4, 6, 8};  
int sum = 0;  
for (auto e : a) sum += e;  
cout << "sum->" << sum << endl;  
- Compiles and runs, but results are undefined.  
sum->20  
- Does not compile. Cannot use range-loop on arrays.  
- Does not compile; e is undefined.  
- sum->8

sum->8



CS150 Final Exam		Study	...
<pre>for (int i = 0; i &lt; 5; i++) for (int j = 0; j &lt; 4; j++) a[j][i] = cnt++; cout &lt;&lt; a[3][2] &lt;&lt; endl;</pre> <ul style="list-style-type: none"><li>- 8</li><li>- 14</li><li>- 9</li><li>- 11</li><li>- 19</li></ul>			
<p>What is true about an uninitialized pointer?</p> <ul style="list-style-type: none"><li>- Dereferencing it is safe, but has no effect.</li><li>- Dereferencing it is undefined behavior</li><li>- Dereferencing it will cause a program crash</li><li>- None of these are true</li><li>- It is set to the nullptr value</li></ul>	Dereferencing it is undefined behavior		
<p>What happens here?</p> <pre>char s[50] = "CS150"; strcat(s, "CS50"); cout &lt;&lt; s &lt;&lt; endl;</pre> <ul style="list-style-type: none"><li>- "CS50"</li><li>- "CS500"</li><li>- Crashes when run.</li><li>- Undefined behavior</li><li>- "CS150CS50"</li></ul>	"CS150CS50"		
<p>The variable *p:</p> <pre>void f(){ int *p = new int{};}</pre> <ul style="list-style-type: none"><li>- stores a memory address</li><li>- is undefined. Code does not compile.</li><li>- is uninitialized</li><li>- stores the value 0 in all versions of C++</li><li>- stores the value 0 in C++11 only</li></ul>	stores the value 0 in C++11 only		
<p>What does this function do?</p> <pre>double mystery(const double a[], size_t len) { double x = 0; for (size_t i = 0; i &lt; len; i++) if (a[i] &lt; x) x = a[i]; return x;}</pre> <ul style="list-style-type: none"><li>- Returns the largest number in the array</li><li>- Does not compile</li><li>- Returns the smallest number in the array</li><li>- Undefined. Depends on the input.</li></ul>	Undefined. Depends on the input.		
<p>Below is a cumulative algorithm using an array and an iterator-based loop. What is printed? (Assume all includes have been added, etc.)</p> <pre>double average(const int <b>beg</b>, <b>const int</b> end){ double sum = 0; size_t count = end - beg; while (beg != end) sum += *beg++; return sum / count;}</pre> <p>int main(){ int a[] = {2, 4, 6, 8}; cout &lt;&lt; average(begin(a), end(a) - 1) &lt;&lt; endl;}</p> <ul style="list-style-type: none"><li>- 4</li><li>- 6</li><li>- Does not compile</li><li>- 5</li><li>- Endless loop when run; likely crashes.</li></ul>	4		

<ul style="list-style-type: none"> <li>- int d[]</li> <li>- int c[2][2];</li> <li>- All of these</li> <li>- int a[][2];</li> </ul>	
<p>The value for the variable b is stored:</p> <pre>int a = 1; void f(int b){ int c = 3; static int d = 4; }</pre> <ul style="list-style-type: none"> <li>- on the stack</li> <li>- on the heap</li> <li>- in the static storage area</li> <li>- in the CPU machine registers</li> <li>- The example does not provide enough information</li> </ul>	on the stack
<p>This code:</p> <pre>void f() { int *p = new int[3]{rand(), rand(), rand()}; if (p[1] != 0 &amp;&amp; p[2] != 0) delete[] p; cout &lt;&lt; p[0] / p[1] / p[2] &lt;&lt; endl; }</pre> <ul style="list-style-type: none"> <li>- has a syntax error</li> <li>- None of these</li> <li>- has a dangling pointer</li> <li>- has a double delete</li> <li>- has a memory leak</li> </ul>	has a dangling pointer
<p>The variable *p:</p> <pre>void f(){ int *p = new int(42); }</pre> <ul style="list-style-type: none"> <li>- stores the value 42 in C++11 only</li> <li>- is uninitialized</li> <li>- stores a memory address</li> <li>- is undefined. Code does not compile.</li> <li>- stores the value 42 in all versions of C++</li> </ul>	stores the value 42 in all versions of C++
<p>What is a common pointer error?</p> <ul style="list-style-type: none"> <li>- Using a pointer without first initializing it</li> <li>- Setting a pointer value to nullptr</li> <li>- You Answered Dereferencing a pointer</li> <li>- Assigning a new value to a pointer</li> <li>- Using indirection on a pointer</li> </ul>	Using a pointer without first initializing it