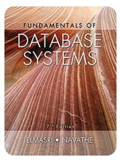
c++ midterm 2 i wanna kms

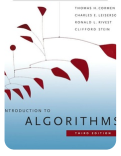
Share

★ Leave the first rating

Textbook solutions for this set



Fundamentals of Database Systems
7th Edition • ISBN: 9780133970777
Ramez Elmasri, Shamkant B. Navathe
687 solutions



Introduction to Algorithms
3rd Edition • ISBN: 9780262033848 (4 more)
Charles E. Leiserson, Clifford Stein, Ronald L. Rivest, Thomas H. Cormen
726 solutions

Search for a textbook or question ›

Terms in this set (133)

What kind of error is this? ex1.cpp:7: undefined reference to 'f()'	linker error
What kind of error is this? ex1.cpp:7:9: warning: missing terminating '"' character a = "hello world'; ^ ex1.cpp:7:9: error: expected expression	syntax error
In a library, the client or test program:	consists of function calls

How many lines of output are printed?
 	
int i = 0;
int j = 0;
while (i < 25){
i = i + 2;
j++ }
cout<< j << endl;
	13

What is the output of the following?
 	
bool token1 = true;
while (token1)
{
for (int i = 0; i < 5; i++)
{
cout << "Hello there" <<
endl;
}
token1 = false;
}
	"Hello there" 5 times

How many times will this display "So far so good"?
 	
int i = 0;
while (i != 15)
{
cout << "So far so good" << endl;
i++;
}
	15 times

int i = 1;
while (i < 20)
{
cout << i << " ";
i = i + 2;
if (i == 15)
{
i = 19;
}
}
	1 3 5 7 9 11 13 19

What kind of error is this? terminate called after throwing an instance of 'std::out_of_range'	runtime error

int i = 0;
while (i != 9)
{
cout << "Loop Execution" << endl;
i++;
}	

What is the output of the following?
 	
int i = 0;
while (i != 9)
{
cout << i << " ";
i = i + 2;
}
	0 2 4 6 8 10 12 14 .... (infinite loop)

What is the output of the following?
 	
int i = 1;
while (i < 10)
{
cout << i << " ";
i = i + 2;
if (i == 5)
{
i = 9;
}
}
	1 3 9

Which prototypes in the following header file contain errors?
 	
#ifndef EXAMPLE_H
#define EXAMPLE_H
#include <string>
 	
string f1(int a);
int f2(double);
void f3(std::string& s, int n);
double f4();
 	
#endif
	f1

Which of these program organization schemes does not work?	Call your functions and define them afterwards
What kind of error is this? ex1.cpp:7:10: error: expected ';' after expression a = 4 ^ ;	syntax error
Which of these documentation tags are used in a function comment?	@param @code @return @endcode
Which of these may go into a header file?	constant definitions function prototypes

Which prototypes in the following header file contain errors?
 	
#ifndef EXAMPLE_H
#define EXAMPLE_H
#include <string>
 	
std::string f1(int a);
int f2(double);
void f3(std::string& s, int n);
double f4();
 	
#endif
	none of these

What is the output of the following?
 	
int i = 1;
while (i != 9)
{
cout << i << " ";
i++;
if (i = 9)
{
cout << "End";
}
}
	1 End

Which of these documentation tags are used in a file comment?	@version @author
A function where an argument is converted to match a parameter	best match
When more than one match is found for the proffered arguments.	ambiguity
A function where each argument is the same type as the corresponding parameter.	exact matches

A group of functions with the same name.	candidate set
A group of functions that have the same name and the correct number of parameters.	viable set
When no match is found for the proffered arguments	empty set
Which of these are ways that functions may be overloaded?	-different order of parameter types -different parameter types -different number of parameters

What prints here?
 	
int i = 5;
while (i) cout << i--;
cout << endl;
	54321

Examine this code. Which is the best prototype?
 	
string s = "dog";
cout << upper(s) << endl;
// DOG
cout << s << endl;
// dog
	string upper(const string&)

To allow f() to change the argument passed here, the parameter str should be declared as:
 	
void f(. . . str);
int main()
{
f("hello");
}
	It's not possible for f() to change the argument passed here

What prints here?
 	
int i = 5;
while (i) cout << --i;
cout << endl;
	4321

Examine this code. Which is the best prototype?
 	
string s = "dog";
upper(s);
cout << s << endl; // DOG
	string upper(const string&)

What prints here?
 	
int i = 5;
while (i--) cout << i;
cout << endl;
	43210

Examine this code. Which is the best prototype? int age; string name = read("Enter your name, age: ", age);	string read(const string&, int&)

What prints here?
 	
int i = 5;
while (i); cout << i--;
cout << endl;
	infinite loop

auto a = '1';
switch (a)
{
case 1: cout << "1"; break;
case 2: cout << "2"; break;
default: cout << "3";
}
cout << endl;	

To allow f() to change the argument passed here, the parameter str should be declared as:
 	
void f(. . . str);
int main()
{
string s = "hello";
f(s);
}
	string&

What prints here?
 	
auto a = 1;
switch (a)
{
case 1: cout << "1";
case 2: cout << "2";
}
cout << endl;
	12

What prints?
 	
string str = "Hello";
for (auto i = 0, len = str.size(); i < len; i++)
cout << str.at(i);
	Does not compile

You can call a single function in several different ways by giving the function ___________:	default arguments

What prints?
 	
void fn(int, double, double&) { cout << "A" << endl; }
void fn(int, int, double&) { cout << "B" << endl; }
void fn(int, int, double) { cout << "C" << endl; }
void fn(int, int, int) { cout << "D" << endl; }
 	
int main()
{
fn(2.5, 1.5, 2.5);
}
	c

Arguments passed to a function that has a constant reference parameter must be:	either lvalues or rvalues

auto a = 1;
switch (a)
{
case 1: cout << "1"; break;
case 2: cout << "2"; break;
default: cout << "3";
}
cout << endl;
	1

What prints here?
 	
auto a = 3, b = 3;
cout << (a != b ? "panda": "tiger") << endl;
	tiger

Default arguments may only be used with reference parameters.	false
If str = "hello", then str.size() > -1.	false
In a do-while loop, (condition) is followed by a semicolon.	true
The compiler determines which overloaded function to call by looking at the type of value the function returns.	false
Append output to a file named z	pwd >> z
Discard both output and errors	rm x > /dev/null 2>&1
Write the output to a new file named z	pwd >> z
Read the input from the file named z	cat < z

Send the output to the input of the program named z	date | z
Has a single char& parameter	get()
Returns the last character read to the input stream	unget()
Examines, but does not read the next character in an input stream	peek()
Replaces the last character read with any character	put()
Called implicitly when an input statement is used as a test condition	fail()
A predicate function	isalpha()
Converts its value argument to a character and sends it to output.	isalpha()

Complete the following code in the lower filter program.
 	
char ch;
while (cin.get(ch))
cout.put(________);
	tolower(ch)

Complete the following code in the echo filter program.
 	
char ch;
while (_________)
cout.put(ch);
	cin.get(ch)

What to process filters do?	-copy files or search for a particular value in a stream (cp and grep) -case modification or changing character order in a stream -stream editing using a sequence of editing commands (sed) -translating data from one form to another (decimal to binary)
Which line runs a.out getting its input from in.txt and appending its output to the file out.txt?	./a.out > in.txt >> out.txt
Assume you have a char variable named ch. How do you "unread" a character already read?	cin.putback(ch);

Assume the user types "brown cow" when this code runs. What is stored in ch2?
 	
char ch1;
auto ch2 = cin.get(ch1);
	cin

Assume the user types "brown cow" when this code runs. What type is ch2?
 	
char ch1;
auto ch2 = cin.get(ch1);
	istream&

Assume the user types "brown cow" when this code runs. What prints?
 	
int n;
if (cin >> n) cout << "X\n";
else cout << "Y\n";
	Y

Assume you have a char variable named ch. How do you read one character from input?	cin.get(ch);
Assume you have a char variable named ch. How do you look ahead before reading a character?	None of these
Which line runs the dom program and sends both output and errors to file named v.data?	./dom > v.data 2>&1

Assume the user types "brown cow" when this code runs. What prints?
 	
char c;
cout << cin.get(c) << endl;
	Does not compile

Which line runs a.out getting its input from in.txt and sending its output to the new file out.txt?	./a.out > out.txt < in.txt

Assume the user types "brown cow" when this code runs. What prints?
 	
char c;
cout.put(cin.get(c));
	Does not compile

Which line runs the prt program and stores its output in a new file named x.data?	./prt > x.data
To test if an I/O operation succeeded you must explicitly call the stream's fail() member function.	false

The expression cin.get(ch) returns a reference to the input stream.	true
You can test if an I/O operation succeeded by explicitly calling the stream's fail() member function.	false
A state filter learns something about the stream by examining characters.	true
Calling cout.put(65) is illegal. Your code will not compile.	false
Create an output file stream object named out and open the text file "expenses.dat", using a single statement.	of stream out("expenses.dat")
Create an output file stream object named out.	ofstream out;

This loop:
 	
char c;
while (c = in.get())
{
cout << c << endl;
}
	is an endless loop

Stream arguments to a function should always be passed:	by reference

What does this code do?
 	
ifstream in("temp.txt");
string x;
int i{0};
while (in >> x) i++;
cout << i << endl;
	counts the number of words in the file

What does this code do?
 	
ifstream in("temp.txt");
char x;
int i{0};
while (in.get(x)) i++;
cout << i << endl;
	counts the number of characters in the file

After opening the input stream in, which of these cannot be used to see if the file was successfully opened?	if (in.opened()) {/ **opened ok** /}

What does this code do?
 	
ifstream in("temp.txt");
char x;
int i{0};
while (in >> x) i++;
cout << i << endl;
	counts the number of non-space characters in the

Assume that the file scores.txt does not exist. What happens?
 	
ofstream out("scores.txt");
out << "Peter" << " " << 20 << endl;
out << "John" << " " << 50 << endl;
	creates a new file, scores.txt and writes two lines of text

Which line opens the file in.txt for reading?	ifstream in("in.txt")
Which of these is not a technique for implementing a sentinel loop?	the counter-controlled pattern
Which line represents the necessary bounds in this loop? 1. string s("Hello CS 150"); 2. while (s.size()) 3. { 4. if (s.at(0) == 'C') break; 5. s = s.substr(1); 6. } 7. cout << s << endl;	2
Which line represents the intentional bounds in this loop? 1. string s("Hello CS 150"); 2. while (s.size()) 3. { 4. if (s.at(0) == 'C') break; 5. s = s.substr(1); 6. } 7. cout << s << endl;	4

ifstream in("temp.txt");
string x;
int i{0};
while (getline(in, x)) i++;
cout << i << endl;	

Read and write characters to memory using streams	sstream
Connect a disk file to an input or output stream.	fstream
Use the predefined stream objects cin and cout	iostream
Determine the category of a character	cctype
Modify the way that memory is converted to characters on input or output	iomanip

Which of the following loop patterns are used here?
 	
int upper = 0;
char ch;
while (in.get(ch))
{
if (ch >= 'A' && ch <= 'Z')
upper++;
}
	data loop, inline test

Stream parameters should always be passed to functions by reference.	true
A loop that reads data until some special value is found is called a sentinel loop.	true
In the loop-and-a-half pattern, you read data before the loop and at the end of the loop.	false
Unformatted I/O means that you read and write data character-by-character.	true
When writing a function with stream parameters, always use the most general type of stream that meets the specification	true
When writing a function with stream parameters, always use the most specific type of stream that meets the specification	false
In the loop-and-a-half, you use Boolean variable to signal when the sentinel is found.	false
The statements that may generate an exception are placed in a ___ block.	try
If no exception is thrown in a try block, all catch blocks associated with that try block are ignored.	true
The try block is followed by one or more ___ blocks.	catch
The heading of a try block can contain ellipses in place of a parameter.	false
Which of the following statements throws a valid exception in C++?	throw 2;
What is the purpose of the throw statement?	It is used to pass control to an error handler when an error situation is detected.
Which of the following blocks is designed to catch any type of exception?	catch(...) { }
In a sequence of try/catch blocks, the last catch block of that sequence should be ___.	catch(. . .) { }

What prints?
 	
string s("hello");
try {
auto x = s.at(s.size());
cout << "one" << endl;
}
catch (const string& e) { cout << "two\n"; }
catch (exception& e) { cout << "three\n"; }
catch (...) { cout << "four\n"; }
	three

Code that may cause an error should be placed in a _________ block and code that handles the error should be inside a _______ block?	try, catch
What statement is used to signal other parts for your program that a particular error has occurred?	throw
What preprocessor directive is not used when you wish to create blocks of code that are only compiled under certain circumstances?	All of these may be used
The C++11 standard library provides the function stoi() to convert a string to an integer. Which library is it found in?	cnvt

template <typename T>
void addem(T a, U b)
{
cout << a << " + " << b << "->"
<< (a + b) << endl;
}	

What term describes this block of code?
 	
#if _APPLE_
istringstream in(" .75");
int n = 3;
in >> n;
#endif
	conditional compilation

Variables tested with the #if preprocessor directive are created using #define.	true
The #if preprocessor directive may compare double literals but not variables.	false
A completion code is a special return value that means "the function failed to execute correctly."	true
The #if preprocessor directive can compare integers.	true
A specialized error handling block of code, is called a catch block.	true
When you throw an exception, control immediately jumps out of the current try block.	true
Calling a function like to_string<int>(3.5) is known as implicit instantiation.	false
You can report a syntax error encountered in your code by using the throw keyword.	false
One of the main problems with the completion code strategy of error handling is that callers can ignore the return value without encountering any warnings.	true
A catch(...) will catch any kind of thrown exception.	true