CS 150 Midterm 2 Study Guide Answers

Share

11 studiers recently ★ Leave the first rating

Terms in this set (221)

An unguarded loop is also known as a test-at-the-bottom loop.	True
Which of these are unguarded loops?	do-while

In a guarded loop, the loop actions are always executed at least once.	False
Match each item with the correct question below. What must I change in the test to go to the next iteration? Can my loop reach its bounds? Has my loop reached its goal? What makes this loop quit?	advance the loop necessary bounds loop postcondition loop bounds
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? 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."	goal
Which of these are indefinite loops?	limit loops, data loops, and sentinel loops

How many times is this loop entered? (That is, how many times is i printed?)
for(int i =0; i<=10; i++)
cout << i;
cout << endl;
	11

How many times is this loop entered? (That is, how many times is i printed?)
 	
for (int i =1; i<10; i++)
cout << i;
cout << endl;
	9

Below is the illustration from the loop building strategy. The highlighted lines represent: While more-characters and current-character not a period	loop bounds
An undeclared error message is a linker error.	False
Which of these documentation tags are used in a function comment?	@code @param

What is the output of the following?
 	
int i = 1;
int sum = 0;
while (i <= 11)
{
sum = sum + i;
i++;
}
cout << "The value of sum is " << sum;
	The value of sum is 66

Match each item with the correct statement below. End a block of source code Required to document functions, global, variables, and constants You name When was it created?	@endcode @file @author @date
A process filter does something to the characters it encounters.	True

In 1735 Leonard Euler proved a remarkable result, which was the solution to the Basel Problem, first posed in 1644 by Pietro Mengoli. This result gave a simple expression for . The formula states that is equal to the limit, as n goes to infinity, of the series . Can this series be computed recursively?	Yes
Unformatted I/O means that you read and write data line-by-line.	False
Infinite recursion can lead to an error known as	stack overflow
In the classic for loop, which portion is used to create the loop control variable?	initialization statement
In the classic for loop, which portion of code is executed after the last statement in the loop body?	update expression
Below is the illustration from the loop building strategy. The highlighted lines represent: Add one to (or increment) the counter variable	goal operation
Below is the illustration from the loop building strategy. The highlighted lines represent: Set counter to 0	goal precondition
In a library, the client or test program:	consists of function calls

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 = 3, b = 3;
cout << (a == b ? "panda": "tiger") << endl;
	panda

What is the output of the following?
 	
string s = "12345";
int i = 1;
while (i < 5)
{
cout << s.substr (i, 1);
i++;
}
	2345

Default arguments appear only in the function prototype.	True

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

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

To test if an I/O operation succeeded you must explicitly call the stream's fail() member function.	False
Calling cout.put(65) prints the character 'A' on output.	True
The getline() function is a member function in the istream class.	False

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

cat < a.txt > b.txt erases the contents of b.txt before writing to it.	True
Below is the illustration from the loop building strategy. The highlighted lines represent: Store the next character from str in current-character	advancing the loop
Match each item with the correct statement below. Actions that occur after the loop is complete Actions occurring inside the loop's body Actions that occur before the loop is encountered A test the determines if the loop should be entered	postcondition operation precondition bounds
A guarded loop is also known as a test-at-the-bottom loop.	False
In the classic for loop, which portion of code is analogous to an if statement?	condition expression
The compiler determines which overloaded function to call by looking at the type of value the function returns.	False
Default arguments may be used with both value and reference parameters.	False
In C++, the standard stream stderr is used to initialize the cout object.	False

Examine the code below.
int mystery3(int n)
{
if (n < 2) return 1;
return n * mystery3(n-1);
}
	mystery3 completes for all inputs
	mystery3 correctly implements the Factorial algorithm

Formatted I/O means that you read and write data token-by-token.	True

Which statement ensures that r() terminates for all values of n?
 	
int mr(int n)
{
// code goes here
return r(n - 1) + n * n;
}
	if (n < 1) { return 1; }

The cin object is an instance of the istream class.	True

This idiomatic pattern is used to count from one value to another.
 	
for(int i=0; i<10; i++)
cout << i;
cout << endl;
	False

A guarded loop is also known as a test-at-the-top loop.	True

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

What is the output of the following?
 	
bool val1 = true;
bool val2 = false;
while (val1)
{
if (val1)
{
cout << "Hello" << endl;
}
val1 = val2;
}
	"Hello" will be displayed only once.

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

When you call a function, the compiler must know:	the number of arguments to pass the name of the function the kind of value returned if any
Which line runs a.out getting its input from in.txt and sending its output to the file out.txt, and its errors to the file err.txt?	./a.out < in.txt > out.txt 2> err.txt
Calling cout.put("A") is illegal. Your code will not compile.	True
In the C++ stream hierarchy, the base class of the stringstream class is:	iostream

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

Which of the following statements is correct about a recursive function?	A recursive function calls itself.

Match each item with the correct statement below. Executable Library File Project File Implementation file	digit-tester libdigits.a makefile digits.cpp

What is the value of r("hello")?
string r(const string& s)
{
if (s.size() < 2) return s;
return s.substr(0, 1) + "*" + r(s.substr(1));
}
	"h**e**l**l**o"

Look at the problem statement below. The _____ of the loop is read a character and increment a counter. "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."	plan

This loop uses asymmetric bounds.
for(int i =1; i<= 10; i++)
cout << i;
cout<<endl;
	False

Match each item with the correct question below. What must I change in the test to go to the next iteration? What must I do to enter the loop? Has my loop reached its goal? Can my loop be entered at all?	advance the loop bounds precondition loop postcondition loop guards
In an unguarded loop, the loop actions are always executed at least once.	True

Match each item with the correct statement below. May not repeat its actions at all Repeats its actions at least once Test for the occurrence of a particular event Conditions under which a loop will repeat its actions	guarded loop unguarded loop indefinite loop loop bounds
An unguarded loop is also known as a test-at-the-top loop.	False
Loop bounds used when searching through input.	sentinel bounds
Loop bounds used when reading files or processing network data.	data bounds

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(1, 2, 3, 4);
}
	Syntax error: no candidates

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

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 (mistake in grammar)

What is the value of r(74757677)?
 	
int r(int n)
{
if (n) return (n % 10 == 7) + r(n / 10); return 0;
}
	5

To use a disk file as a data stream source or sink, use the <ifstream> header	False
The Unix filter to use for searching through text to find a particular word is called find.	False

What does this function do?
 	
int mystery(int n)
{
if (n == 1) return 1;
return n * mystery(n+1);
}
	Produces a stack overflow

In 1735 Leonard Euler proved a remarkable result, which was the solution to the Basel Problem, first posed in 1644 by Pietro Mengoli. This result gave a simple expression for . The formula states that is equal to the limit, as n goes to infinity, of the series . Which statement below is the recursive case for a recursive implementation that approximates this infinite series?	return 1.0 / (number * number) + computePI(number - 1);
A process filter learns something about the stream by examining characters.	False

In a guarded loop, the loop actions may never be executed.	true
The highlighted section below illustrates: current-character not a period	intentional condition

How many times is this loop entered? (That is, how many times is i printed?)
for (int i = 1, i <= 10; i++)
cout << i;
cout << endl;
	10

Using the loop-building strategy from the lessons, which of these are part of the loop mechanics?	advancing the loop loop bounds bounds precondition

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

Default arguments allow you to write several different functions that have the same name.	false
Function overloading lets you call a single function in several different ways.	False

Which of these prototypes is the best one to use in this circumstance?
int main()
{
string str{"TO BE OR NOT TO BE"};
properCase(str);
cout << str << endl;
}
	void properCase(string&)

What kind of error is this? ex1.cpp:6:12: error: no viable conversion from 'int' to 'string' string a = 15; ^ ~~	Type error (wrong initialization or assignment)
The C++ term for what is called a superclass in other languages is base class.	True

In the C++ stream hierarchy, base class of the istream class is:	ios
Which of these are not process filters?	counting word transitions print one sentence per line compress input by turning off echo when reading blank spaces

One remarkably simple formula for calculating the value of is the so-called Madhava-Leibniz series: pi = 1 - 1/3 + 1/5 - 1/7 + 1/9-.... Consider the recursive function below to calculate this formula:
 	
double computePI(int number)
{
if (number <= 1) { return 1.0;}
int oddnum = 2 * number - 1;
return computesign(number) * 1.0 / oddnum + computePI(number - 1);
}
 	
In this recursive function, what is the recursive base case?
	When the parameter variable is less than or equal to one

What is the value of r("axxbxx")?
string r(const string& s)
{
auto front = s.substr(0, 1);
if (front.empty()) return "";
return (front == "x" ? front : "") + r(s.substr(1));
}
	"a b "

At the lowest level, all input and output is a stream of bytes flowing through your program.	True
You can test if an I/O operation succeeded by explicitly calling the stream's fail() member function.	True
In the C++ stream hierarchy, the base class of the ostream class is:	ios
Below is the illustration from the loop building strategy. The highlighted lines represent: If current-character is a period then add one to the counter to account for the period. else set counter to -2	loop postcondition

This idiomatic pattern is used to count from one value to another.
 	
for (int i = 1; i <= 10; i++)
cout << i;
cout << endl;
	True

Match each item with the correct question below. What must I change in the test to go to the next iteration? What must I do to enter the loop? Has my loop reached its goal? Can be loop be entered at all?	advance the loop bounds precondition loop precondition loop guards

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

Loops that do some processing and then compare the results against a boundary condition are called __________?	loop limits
A tool named Doxygen is often used to generate HTML user docs from C++ code.	True
Match each item with the correct statement below. Meaning of value returned from a function Begin a block of source code Information about the library name and meaning for a paramter	@return @code @version @param
To use a disk file as a data stream source or sink, use the <fstream> header	true

What Java and other OO languages call a superclass, C++ calls a	base class
A filter program, like cat, is designed to be run interactively from the terminal.	false

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

Assume you have a char variable named ch. How do you look ahead before reading a character?	cin.peek();

What is the value of r(12777)?
 	
int r(int n)
{
if (0 == n) return 0;
int x = n % 10 == 7; // 0 or 1
return x + r(n / 10);
}
	3

What changes about this function if lines 4 and 5 are swapped? 1. void myfun(const string& word) 2. { 3. if (word.size() == 0) { return; } 4. myfun(word.substr(1)); 5. cout << word[0]; 6. }	reverses the order in which the characters of the string are printed

Examine the code below and match the statements following it.
 	
int mystery3(int n) {
if (n < 2) return 1;
return n * mystery3(n - 1);
}
 	
The algorithm implemented is:
 	
mystery3 completes for all inputs:
 	
mystery3 is a recursive wrapper:
 	
mytery3 returns the correct answer for all inputs:
	factorial
	
	true
	
	false
	
	false

How many lines of output are printed?

```
int i = 0;
int j = 0;
while (i < 25)
{
i = i + 2;
j++;
}
cout << j << endl;
```

13

Match the following code the the answers below

```
template <typename T, typename U>
U pickle(T& a, const U& b) {
a += b;
return b;
}

int main()
{
int x = 42;
auto a = pickle(x, 4.5);
cout << a << endl;
cout << x << endl;
}
```

Inside main, the variable a is type:

inside main, the value printed for a is:

inside maaun, the value printed for x is:

}

double

4.5

46

What prints?

```
string str = "Hello";
for (auto i = 0, len = str.size(); i < len; i++)
cout << str.at(i);
```

Hell

What kind of error is this?

```
~/workspace/$./ex1
The Patriots won the 2018 superbowl
```

None of these (THIS ISN'T AN ERROR)

Which command sorts the lines in input.txt and stores the sorted output in a new file named sorted.txt?

```
cat < input.txt | sort > sorted.txt
```

Given the overloaded functions prototypes and the variable definition below, which of the function calls will fail to compile?
 	
int f(int&);
int f(int);
int f(int, int);
int a = 7;
	f(a)

Given the overloaded functions prototypes and the variable definition below, which of the function calls will fail to compile?
 	
int f(int&);
int f(const int&);
int f(int, int);
int a = 7;
	None of these fail to compile

Assume that the input is 4 4 3 2 5. What will print?
int i = 1;
int n;
cin >> n;
do
{
i++;
cin >> n;
}
while (n % 2);
cout << i << endl;
	2

Assume that the input is 5 5 4 3 5. What will print?
int i = 1;
int n;
do
{
cin >> n;
i++;
}
while (n % 2);
cout << i << endl;
	4

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?
 	
string str = "Hello";
for (int i = str.size() - 1; i >= 0; i--)
cout << str.at(i);
	olleH

What prints?
 	
string str = "Hello";
for (size_t i = str.size() - 1; i >= 0; i--)
cout << str.at(i);
	Crashes when run

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

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

Which prototypes in the following header filer 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

[1] What must I change in the test to go to the next iteration? [2] What information is produced? [3] What must I do to enter the loop? [4] Can my loop reach its bounds? [5] Has my loop reached its goal? [6] How is the data processed? [7] Can my loop be entered at all? [8] What makes this loop quit?	[1] advance the loop [2] goal precondition [3] bounds precondition [4] necessary bounds [5] loop postcondition [6] loop operations and actions [7] loop guards [8] loop bounds

[1] May not repeat its actions at all [2] Keeps processing input until a particular value is found in input. [3] Repeats its actions at least once [4] Keeps processing until the output gets no closer to the answer. [5] Test for the occurrence of a particular event [6] Repeats its actions a fixed number of times [7] Conditions under which a loop will repeat its actions [8] Keeps processing until the input device signals that it is finished.	[1] guarded loop [2] sentinel loop [3] unguarded loop [4] limit loop [5] indefinite loop [6] definite loop [7] loop bounds [8] data loop
[1] Actions that occur after the loop is complete [2] Actions occuring inside the loop's body [3] Actions that occur before the loop is encountered [4] A test that determines if the loop should be entered	[1] postcondition [2] operation [3] precondition [4] bounds

Which of these is a flow-of-control statement?	for (auto e : s) ...
	if (x < 3) ... else ...
	while (x < 3) ...

Which of these are guarded loops?	for
	while

Which are the two major categories of loops?	definite indefinite
Using the loop-building strategy from Chapter 5, which of these are part of the loop mechanics?	loop bounds bounds precondition advancing the loop
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."	bounds
Loop bounds often used in scientific and mathematical applications.	limit bounds
In the classic for loop, loop control variables going from 0 to less-than n are said to employ:	asymmetic bounds

How many times is this loop entered? (That is, how many times is i printed?)
for (int i = 0; i < 10; i++)
cout << i;
cout << endl;
	10

In the classic for loop, which portion of code is not followed by a semicolon?	update expression
Below is the illustration from the loop building strategy in Chapter 5. The highlighted lines represents: If the variable str has any characters then	a loop guard
Below is the illustration from the loop building strategy in Chapter 5. The highlighted lines represents: Create the variable current-character as a characterPlace the first character in str into current-character	bounds precondition
The highlighted selection below illustrates: While more-characters	a necessary condition
The highlighted selection below illustrates: current-character not a period	an intentional condition
Below is the illustration from the loop building strategy in Chapter 5. The highlighted lines represents: If current-character is a period then	loop postcondition
In an unguarded loop, the loop actions may never be executed.	False
Loops are used to implement iteration in C++.	True
Loops are used to implement selection in C++.	False

This loop uses asymmetric bounds.
for (int i = 1; i < 10; i++)
cout << i;
cout << endl;
	True

This loop uses asymmetric bounds.
for (int i = 0; i < 10; i++)
cout << i;
cout << endl;
	True

End a block of source code Meaning of value returned from a function Required to document functions, global variables and constants Begin a block of source code Your name Information about the library When was it created? Name and meaning for a parameter	@endcode @return @file @code @author @version @date @param
Which of these documentation tags are used in a file comment?	@version @author @date @file
What kind of error is this? ex1.cpp:7:10: error: expected ';' after expression a = 4 ^ ;	Syntax error (mistake in grammar)
What kind of error is this? ex1.cpp:6:5: error: use of undeclared identifier 'a' a = 4; ^	Compiler error (something is missing when compiling)

What is the output of the following?
string s = "abcde";
int i = 1;
while (i < 5)
{
cout << s.substr (i, 1);
i++;
}
	bcde

What is the output of the following?
 	
int i = 1;
while (i <= 10)
{
cout << "Inside the while loop" << endl;
i = i * 11;
}
	"Inside the while loop" will be displayed only once.

What is the output of the following?
 	
int i = 1;
int sum = 0;
while (i <= 13)
{
sum = sum + i;
i = i + 3;
}
cout << "The value of sum is " << sum;
	The value of sum is 35

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

What is the output of the following?
 	
int i = 0, j = 0;
while (i < 125)
{
i = i + 2;
j++;
}
cout << j << endl;
	63

Header files must explicitly qualify each name from the standard library with std::	True
Header files may use the statement using namespace std;	False
An undefined error message is a linker error.	True
An undefined error message is a compiler error	False
An undeclared error message is a run-time error	False

Implementation files may use the statement using namespace std; Implementation files must explicitly qualify each name from the standard library with std::	True False
Parameter names are optional in the function prototype	True
Parameter names are optional in the function definition	False

How many lines of output are printed?
int i = 0;
while (i != 9)
{
cout << "Loop Execution" << endl;
i++;
}
	9

How many lines of output are printed?
int count = 0;
while (count != 9)
{
cout << "Monster Mash" << endl;
if ((count % 2) == 0) {count++;}
else{count--;}
}
	Infinite

What is the output of the following?
bool token = false;
while (token)
{
cout << "Hello World!" << endl;
}
	No Output

Which line in the function "skeleton" below contains an error?
#include "digits.h" // 1.
int firstDigit(int n); // 2.
{ // 3.
return 0; // 4.
} // 5.
	// 2.

Which line in the function "skeleton" below contains an error?
 	
#include "borgia.h" // 1.
void primoTiara(int n) // 2.
{ // 3.
return 0; // 4.
} // 5.
	// 4.

Which line in the function "skeleton" below contains an error?
 	
#include "digits.h" // 1.
int firstDigit(int n) // 2.
{ // 3.
return 0; // 4.
}
	None of these

What kind of error is this? ex1.cpp:7: undefined reference to ` f()'	Linker error (something is missing when linking)
What kind of error is this? terminate called after throwing an instance of 'std::out_of_range'	Runtime error (throws exception when running)
What kind of error is this? Segmentation fault	Operating system signal or trap
In a library, the implementation file:	consists of function definitions
In a library, the interface file:	consists of declarations or prototypes
In a library, the makefile:	consists of instructions that produce the executable
In a client file you should compare your function's value to the ________?	expected value
In a client file, the value returned from calling your function is the________?	actual value
An incomplete, yet compilable, linkable and executable function is called a _______ ?	stub
Which of these program organization schemes does not work?	Call your functions and define them afterwards.
Which of these may go into a header file?	function prototypes constant definitions

Header guards:	end with the directive #endif includes the directive #define go in every interface file start with the directive #ifndef
Executable	digit-tester
Object file	digits.o
Library file	libdigits.a
Interface file	digits.h
Project file	makefile
Client file	digit tester.cpp
Implementation file	digits.cpp
What Java and other OO languages call a subclass, C++ calls a _______.	derived class

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()
{
auto n = 3.5;fn(1, 2.5, n);
}
	A

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

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(1, 2, 3.5);
}
```

C

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, 7);
}
```

D

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()
{
auto n = 3.5;fn(1, 2, n);
}
```

Syntax error: ambiguous

What prints here?

```
auto a = 4, b = 3;
cout << (a == b ? "panda": a % 2 ? "stork": "tiger")
<< endl;
```

Tiger

What prints here?

```
auto a = 3, b = 3;
cout << (a != b ? "panda": a % 2 ? "stork": "tiger")
<< endl;
```

Stork

What prints here?
 	
auto a = 3, b = 3;
cout << a == b ? "panda" : "tiger" << endl;
	Does not compile

Function overloading allows you to write several different functions that have the same name.	True
Overloaded functions have the same name but different parameter types.	True
Overloaded functions have the same name but different parameter names.	False
In a while loop, (condition) is followed by a semicolon.	False
A while loop is a hasty or unguarded loop.	False

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

What prints here?
 	
auto a = 1;switch (a)
{
case 1: cout << "1";
case 2: cout << "2";
case 3:
}
cout << endl;
	Does not compile

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

What prints here?
 	
auto a = 'A';switch (a)
{
case 64: cout << "?";
case 65: cout << "A";
case 66: cout << "B";
}
cout << endl;
	AB

The compiler determines which overloaded function to call by looking at the number, types and order of the arguments passed to the function.	True
Default arguments let you call a single function in several different ways.	True
Default arguments may only be used with value parameters.	True
Default arguments may only be used with reference parameters.	False
Default arguments appear only in the function implementation.	False
Fatal error messages should be printed to cerr.	True
Fatal error messages should be printed to cout.	False
Calling break() terminates a program immediately and passes an error code back to the operating system.	False
If str = "hello", then str.size() > -1.	False
Calling exit() terminates a program immediately and passes an error code back to the operating system.	True

A parameter with a default argument cannot appear before a parameter without a default argument.	True
A do-while loop is also called a hasty loop.	True
In a do-while loop, (condition) is followed by a semicolon.	True
Which line runs the dd program and sends its errors to the file named z.data?	./dd 2> z.data

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

Two quantities a and b are said to be in the golden ratio if (a + b) / a is equal to a / b. Assuming a and b are line segments, the golden section is a line segment divided according to the golden ratio: The total length (a + b) is to the longer segment a as a is to the shorter segment b. One way to calculate the golden ratio is through the continued square root (also called an infinite surd): golden ratio = sqrt (1 + sqrt(1 + sqrt(1....... If the function double golden (int) is a recursive implementation of this function, what should be the recursive call in that function?	return sqrt (1.0 + golden(number - 1)

What does this function do?
int mystery(int n, int m)
{
if (n == 0) return m;
return m * 10 + mystery(n / 10) + n % 10;
}
	computes the reverse of the input n

What does this function do?
int mystery(int n)
{
if (n == 1) return 1;
return n + mystery(n-1);
}
	Computes the Gauss series (sum) of 1..n

What does this function do?
int mystery(int n)
{
if (n == 1) return 1;
return n * mystery(n+1);
}
	produces a stack overflow

Which line advances the 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	5
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