Midterm 2 Study Guide

Due No due date **Points** 25 **Questions** 25 **Time Limit** 30 Minutes **Allowed Attempts** Unlimited

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 7	28 minutes	21 out of 25
LATEST	Attempt 10	30 minutes	20.17 out of 25
	Attempt 9	29 minutes	20 out of 25
	Attempt 8	29 minutes	20.5 out of 25
	Attempt 7	28 minutes	21 out of 25
	Attempt 6	24 minutes	21 out of 25
	Attempt 5	20 minutes	17.83 out of 25
	Attempt 4	30 minutes	17.67 out of 25
	Attempt 3	22 minutes	16 out of 25
	Attempt 2	16 minutes	16.17 out of 25
	Attempt 1	30 minutes	18.33 out of 25

Correct answers are hidden.

Submitted Jun 28 at 3:11am

Question 1

1 / 1 pts

Match each item with the correct statement below.

Keeps processing input until a particular value is found in input.	sentinel loop
Keeps processing until the output gets no closer to the answer.	limit loop
Repeats its actions a fixed number of times	definite loop
Keeps processing until the input device signals that it is finished.	data loop

Question 2

1 / 1 pts

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
- ( ) 4
- ( ) None of these
- ( ) 2

Question 3

1 / 1 pts

Match each item with the correct statement below.

Actions that occur after the loop is complete	postcondition
Actions occurring inside the loop's body	operation
Actions that occur before the loop is encountered	precondition
A test the determines if the loop should be entered	bounds

Incorrect

Question 4

0 / 1 pts

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.*

- plan
- goal
- bounds
- None of these

Question 5

1 / 1 pts

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
- 9
- Never
- 11

Question 6

1 / 1 pts

Below is the illustration from the loop building strategy. The ***highlighted lines*** represent:

```
Given: the variable str is a string (may be empty)
Create the counter variable, initialized to -1
If the variable str has any characters then
{
    Set counter to 0
    Create the variable current-character as a character
    Place the first character in str into current-character
    While more-characters and current-character not a period
    {
       Add one to (or increment) the counter variable
       Store the next character from str in current-character
    }
    If current-character is a period then
       Add one to the counter to account for the period.
    Else
      Set counter to -2
}
If counter is -1 the string was empty
Else if counter is -2 there was no period
```

- ○ goal precondition
- ○ loop bounds
- ○ loop postcondition
- ◉ advancing the loop
- ○ goal operation
- ○ bounds precondition

Question 7

1 / 1 pts

In the classic *for* loop, which portion of code is executed ***after the last statement in the loop body***?

- ○ None of these
- ◉ update expression
- ○ assignment statement
- ○ condition expression
- ○ first statement following the loop
- ○ initialization statement

Incorrect

Question 8

0 / 1 pts

This idiomatic pattern is used to count from one value to another.

```
for (int i = 0; i < 10; i++)
    cout << i;
cout << endl;
```

- ◉ True
- ○ False

Question 9

1 / 1 pts

What prints?

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

- Hell
- Hello
- Crashes when run
- (x) Does not compile
- Undefined behavior

Partial

Question 10

0.5 / 1 pts

Which of these documentation tags are used in a ***function comment?***

- [ ] `@code`
- [ ] `@file`
- [ ] `@version`
- [x] `@param`

Question 11

1 / 1 pts

Implementation files may use the statement `using namespace std;`

- (x) True
- False

Question 12

1 / 1 pts

What kind of error is this?

```
ex1.cpp:6:5: error: use of undeclared identifier 'a'
    a = 4;
    ^
```

- None of these
- Syntax error (mistake in grammar)
- Linker error (something is missing when linking)
- Runtime error (throws exception when running)
- (x) Compiler error (something is missing when compiling)
- Operating system signal or trap
- Type error (wrong initialization or assignment)

Question 13

1 / 1 pts

What is the output of the following?

```
int i = 0;
while (i != 9)
{
    cout << i << " ";
```

```
    i = i + 2;
}
```

- `10 12 14 16 18 ….` (infinite loop)
- `0 2 4 6 8`
- No output
- `0 2 4 6 8 10 12 14 ….` (infinite loop)

Question 14

1 / 1 pts

What prints here?

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

- Undefined behavior
- `tiger`
- Does not compile
- Crashes when run
- `panda`

Question 15

1 / 1 pts

An ***undefined*** error message is a linker error.

- True
- False

Question 16

1 / 1 pts

What prints here?

```
int i = 5;
while (i) cout << --i;
cout << endl;
```

- `54321`
- `43210`
- Syntax error: `i` is not a Boolean expression
- Infinite loop
- `4321`

Question 17

1 / 1 pts

What is the output of the following?

```
bool token = false;
while (token)
```

```
{
    cout << "Hello World!" << endl;
}
```

- (●) No output
- ( ) Hello World!
- ( ) Hello World! will be displayed infinitely many times
- ( ) No output because of compilation error

Question 18

1 / 1 pts

If an output stream's file is missing when you try to open it, its `fail()` member function returns `false`.

- (●) True
- ( ) False

Incorrect

Question 19

0 / 1 pts

Two quantities *a* and *b* are said to be in the *golden ratio* if $\frac{(a+b)}{a}$ is equal to $\frac{a}{b}$. 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 = $\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\ldots}}}}$. In a recursive implementation of this function, what should be the *base case* for the recursion?

- (●) `if (number <= 1) { return sqrt(number);}`
- ( ) `if (number <= 1) { return pow(number, 2.0);}`
- ( ) `if (number <= 1) { return 1.0;}`
- ( ) `if (number <= 1) { return 0.0;}`

Incorrect

Question 20

0 / 1 pts

Assume the user types "`brown cow`" when this code runs. What prints?

```
char c;
cout << cin.get(c) << endl;
```

- ( ) `brown cow`
- (●) `true (or 1)`
- ( ) Does not compile
- ( ) `b`

Question 21

1 / 1 pts

In C++, the standard stream `stderr` is used to initialize the `cout` object.

- ○ True
- ◉ False

Question 22

1 / 1 pts

Examine the code below:

```
int mystery3(int n) {
    if (n < 2) return 1;
    return n * mystery3(n - 1);
}
```

- ☑ In mysterm3, if (n < 2) is a base case
- ☑ mystery3 correctly implements the Fibonacci algorithm
- ☑ mystery3 is efficient
- ☐ mystery3 returns the correct answer for all inputs

mystery3 is an implementation of the Factorial algorithm. It completes for all inputs, but negative inputs produce the wrong output. It is efficient and it is not a wrapper. if (n < 2) is a base case.

Question 23

1 / 1 pts

The redirection pipe symbol is a pair of vertical bars (| |).

- ○ True
- ◉ False

Question 24

1 / 1 pts

This loop:

```
char c;
while (in.get(c))
{
    cout << c << endl;
}
```

- ○ is an endless loop
- ○ illustrates token-based stream processing
- ○ illustrates line-based stream processing
- ○ has a syntax error
- ◉ illustrates raw character I/O

Partial

Question 25

0.67 / 1 pts

Which of the following symbol(s) can be used to redirect the output to a file or another program?

- [ ] <
- [ ] >>
- [x] >
- [ ] <<
- [x] |