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 11	29 minutes	25 out of 25
LATEST	Attempt 19	22 minutes	20 out of 25
	Attempt 18	30 minutes	23 out of 25
	Attempt 17	19 minutes	24 out of 25
	Attempt 16	21 minutes	24 out of 25
	Attempt 15	30 minutes	23 out of 25
	Attempt 14	22 minutes	24 out of 25
	Attempt 13	25 minutes	24 out of 25
	Attempt 12	30 minutes	22 out of 25
	Attempt 11	29 minutes	25 out of 25
	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 29 at 4:23pm

Question 1

1 / 1 pts

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
- 11
- Never

10

Question 2

1 / 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.*

goal

plan

bounds

None of these

Question 3

1 / 1 pts

Below is the illustration from the loop building strategy. The ***highlighted lines*** represent. Store the next character from str in current-character:

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

Incorrect

Question 4

0 / 1 pts

This loop uses asymmetric bounds.

```
for (int i = 1; i <= 10; i++)
    cout << i;
cout << endl;
```

- ◉ True
- ○ False

Question 5

1 / 1 pts

Below is the illustration from the loop building strategy. The ***highlighted lines*** represent. Create the variable current-chracter as a character:

```
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
```

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

Question 6

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;
```

- ○ Never
- ○ 10
- ○ 9
- ◉ 11

Question 7

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;
```

- Never
- 10
- 9
- 11

Question 8

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

Question 9

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;
```

- None of these
- 2
- 4
- 5

Question 10

1 / 1 pts

The compiler determines which overloaded function to call by looking at the number, types and order of the arguments passed to the function.

- [x] True
- [ ] False

Question 11

1 / 1 pts

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

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

Question 12

1 / 1 pts

What prints here?

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

- [ ] Does not compile
- [ ] panda
- [x] tiger
- [ ] Undefined behavior
- [ ] Crashes when run

Incorrect

Question 13

0 / 1 pts

To allow *f()* to change the argument passed here, the parameter *str* should be declared as:

```
void f( . . . str);
int main()
{
    f("hello");
}
```

- [x] string&
- [ ] const string&

- const string
- It is not possible for *f()* to change the argument passed here.
- string

Question 14

1 / 1 pts

Examine the following variables and function calls
Match each item with the correct statement below.

```
int able = 3;
int baker = f1(able);
cout << able << baker << endl; // 64

string charlie;
f2("hello", charlie);
cout << charlie << endl; // Hello Carl
```

Returned value	baker
Output argument (parameter)	charlie
Input argument (parameter)	hello
Input/output argument (parameter)	able

Incorrect

Question 15

0 / 1 pts

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);
- [ ] f(2.0);
- [x] None of these fail to compile
- [ ] f('a', 'b')
- [ ] f(3)

Question 16

1 / 1 pts

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);
}
```

- [x] C
- [ ] D
- [ ] B
- [ ] A
- [ ] Syntax error: no candidates
- [ ] Syntax error: ambiguous

Question 17

1 / 1 pts

An incomplete, yet compilable, linkable and executable function is called a _____ ?

- [ ] prototype
- [ ] declaration
- [x] stub
- [ ] None of these

Question 18

1 / 1 pts

Unformatted I/O means that you read and write data line-by-line.

- [ ] True
- [x] False

Question 19

1 / 1 pts

Assuming that you need to write a recursive function `calc_prod(int n)` to calculate the product of the first n integers, which of the following would be a correct way to simplify the input for the recursive call?

- ○ Call `calc_prod(n - 2)` and multiply by n.
- ○ Call `calc_prod(1)` and multiply by n.
- ◉ Call `calc_prod(n - 1)` and multiply by n.
- ○ Call `calc_prod(n + 1)` and multiply by n.

Question 20

1 / 1 pts

Examine the code below and match the statements following it.

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

mystery3 has a stack overflow for some numbers.	False
mystery3 correctly implements its algorithm	True
if (n < 2) is a . . .	base case
mystery3 is efficient	True

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.

Incorrect

Question 21

0 / 1 pts

This loop:

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

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

Question 22

1 / 1 pts

Which line runs the `dwk` program and gets its input from a file named `y.data`?

- (•) `./dwk < y.data`
- `None of these`
- `./dwk >> y.data`
- `./dwk > y.data`
- `./dwk | y.data`
- `./dwk << y.data`

Question 23

1 / 1 pts

At the lowest level, all input and output is a stream of bytes flowing through your program.

- (•) True
- False

Incorrect

Question 24

0 / 1 pts

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));
}
```

- Stack overflow
- "xxxx"

- "ab"
- "ax bx "
- "a b "

Question 25

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