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 6	24 minutes	21 out of 25
LATEST	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

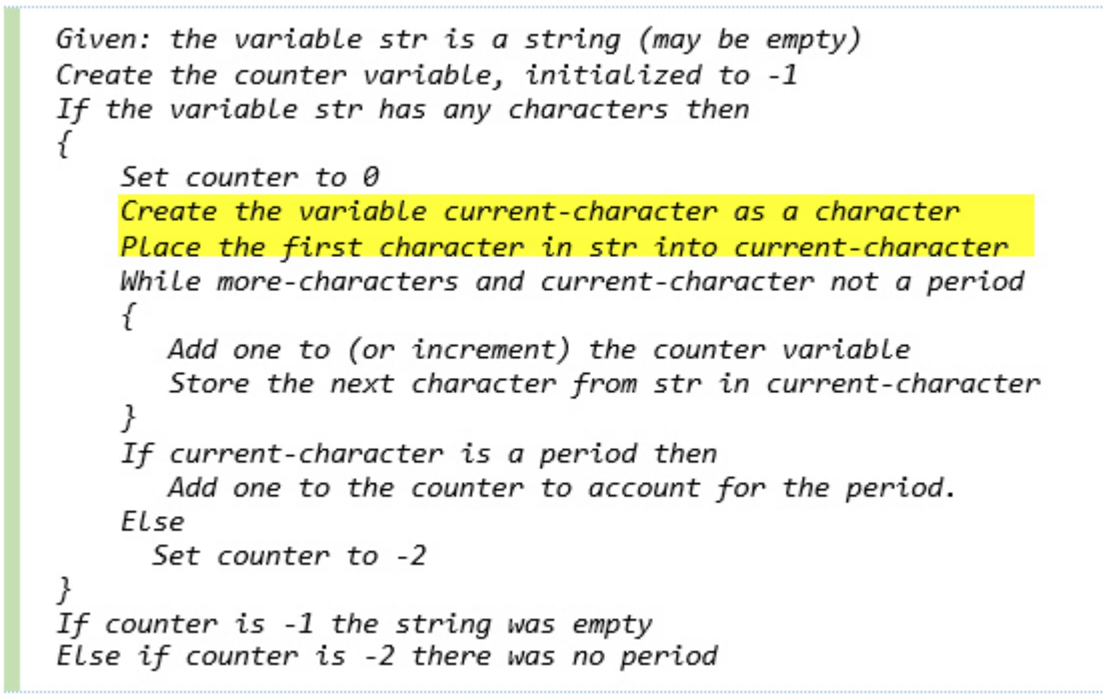
ⓘ Correct answers are hidden.

Submitted Jun 27 at 11:13pm

Question 1

1 / 1 pts

Below is the illustration from the loop building strategy. The ***highlighted lines*** represent. Create the variable current-character 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
```

○ goal precondition

○ advancing the loop

● bounds precondition

○ loop bounds

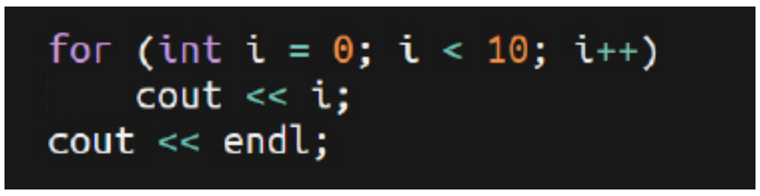
○ goal operation

○ loop postcondition

Question 2

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

○ 11

○ 9

Question 3

1 / 1 pts

Loop bounds often used in scientific and mathematical applications.

● limit bounds

○ data bounds

○ None of these

○ sentinel bounds

Question 4

1 / 1 pts

Which of these are ***unguarded*** loops?

☑ do-while

☐ if

☐ while

☐ for

☐ if-else

Question 5

1 / 1 pts

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

□ goal precondition

☑ advancing the loop

☑ loop bounds

☑ bounds precondition

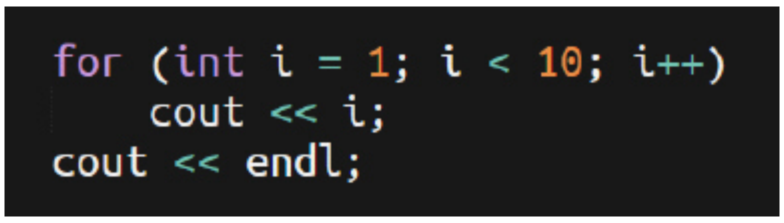
□ post condition

□ 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 = 1; i < 10; i++)
    cout << i;
cout << endl;
```

○ 10

○ Never

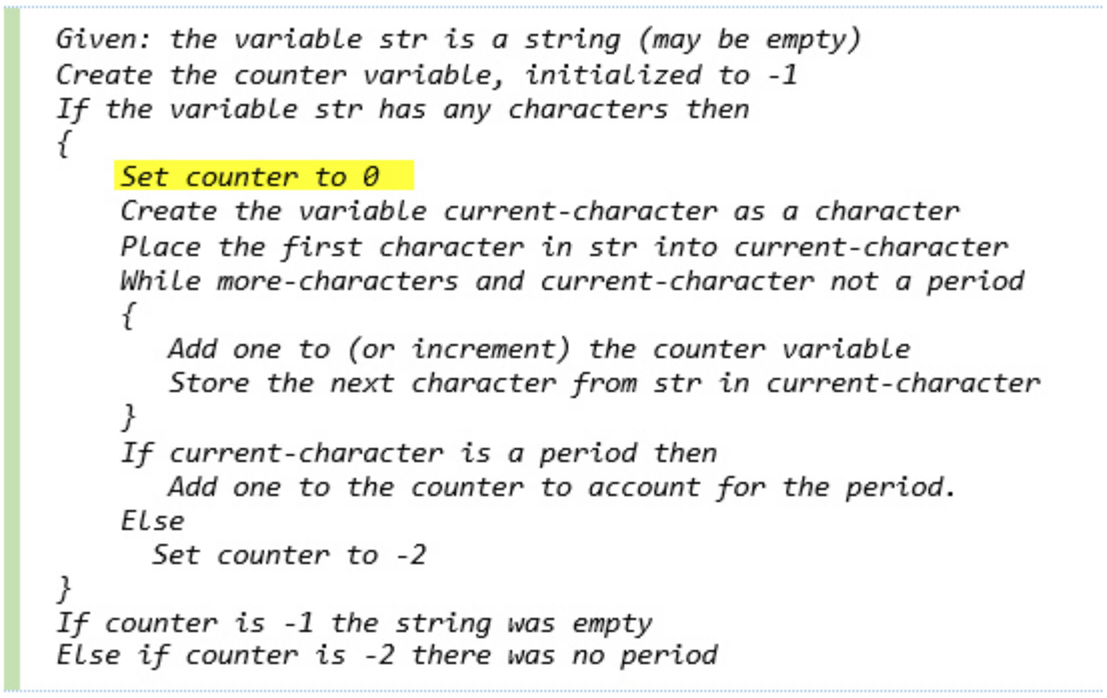
○ 11

● 9

Question 7

1 / 1 pts

Below is the illustration from the loop building strategy. The ***highlighted lines*** represent.
Set counter to 0:



```
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 postcondition

○ bounds precondition

○ advancing the loop

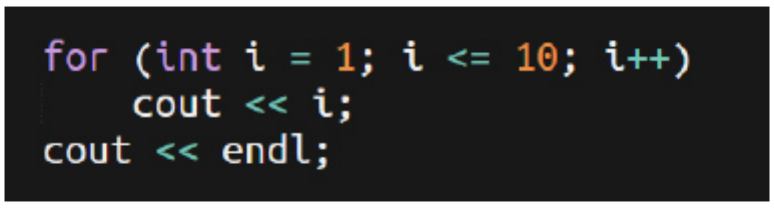
○ goal operation

○ loop bounds

Question 8

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;
```

○ 11

○ 9

○ Never

● 10

Question 9

1 / 1 pts

In a ***guarded*** loop, the loop actions are always executed at least once.

○ True

● False

Question 10

1 / 1 pts

What is the output of the following?

```
bool token = false;
while (token)
{
    cout << "Hello World!" << endl;
}
```

○ Hello World! will be displayed infinitely many times

○ No output because of compilation error

○ Hello World!

● No output

Incorrect

Question 11

0 / 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()
{
    auto n = 3.5;
    fn(1, 2, n);
}
```

○ Syntax error: no candidates

○ C

● B

○ A

○ D

○ Syntax error: ambiguous

Question 12

1 / 1 pts

Default arguments appear only in the function implementation.

○ True

● False

Question 13

1 / 1 pts

If a prototype in a header file has a parameter that is a library type, the header file must `#include` the appropriate library header.

● True

○ False

Question 14

1 / 1 pts

An ***undeclared*** error message is a runtime error.

○ True

● False

Incorrect

Question 15

0 / 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, 7);
}
```

● Syntax error: no candidates

○ A

○ B

○ Syntax error: ambiguous

○ C

○ D

Question 16

1 / 1 pts

An ***undeclared*** error message is a compiler error.

● True

○ False

Question 17

1 / 1 pts

What kind of error is this?

```
terminate called after throwing an instance of 'std::out_of_range'
```

○ Type error (wrong initialization or assignment)

○ Linker error (something is missing when linking)

○ Syntax error (mistake in grammar)

○ None of these

○ Operating system signal or trap

● Runtime error (throws exception when running)

○ Compiler error (something is missing when compiling)

Question 18

1 / 1 pts

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 digits in the file

○ Counts the number of words in the file

○ Gets stuck in an endless loop

○ Counts the number of lines in the file

○ Counts the number of non-space characters in the file

● Counts the number of characters in the file

Question 19

1 / 1 pts

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

● True

○ False

Question 20

1 / 1 pts

Assume you have a `char` variable named `ch`. How do you look ahead before reading a character?

○ `cin.unget(ch);`

○ `None of these`

● `cin.peek();`

○ `cin.get(ch);`

○ `cin.seek(ch);`

Question 21

1 / 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));
}
```

○ `"xxxx"`

○ Stack overflow

○ `"ax bx "`

○ `"a  b  "`

● `"ab"`

Incorrect

Question 22

0 / 1 pts

The file `expenses.txt` contains the line: `Hotel, 3 nights. $ 1,750.25`. What prints?

```
ifstream in("expenses.txt");
char c;
while (in.get(c))
{
    if (isdigit(c)) {
        in.unget();
        double n;
        in >> n;
        cout << n << 'x';
    }
}
```

○ None of these

○ `3x1x750.25x`

○ `3x (then cin fails)`

○ `3x1x7x5x0x2x5x`

● `3x1x750x25x`

○ `3x1x750x25x`

Question 23

1 / 1 pts

Programs that process streams of characters are called text _______________.

○ accessors

○ mappers

○ reducers

● filters

○ mutators

Incorrect

Question 24

0 / 1 pts

What is the value of ***`r("hello")`***?

```
string r(const string& s)
{
    if (s.size() > 1) {
        string t = s[0] == s[1] ? "" : "*";
        return t + s[0] + r(s.substr(1));
    }
    return s;
}
```

○ `"h*e*ll*o"`

○ Stack overflow

○ `"*h*el*lo"`

○ `"h*e*ll*o*"`

● `"hel*lo"`

Question 25

1 / 1 pts

When writing a function with stream parameters, always use the **most specific** type of stream that meets the specification.

○ True

● False