Midterm 2 Study Guide Results for Albert Nguyen

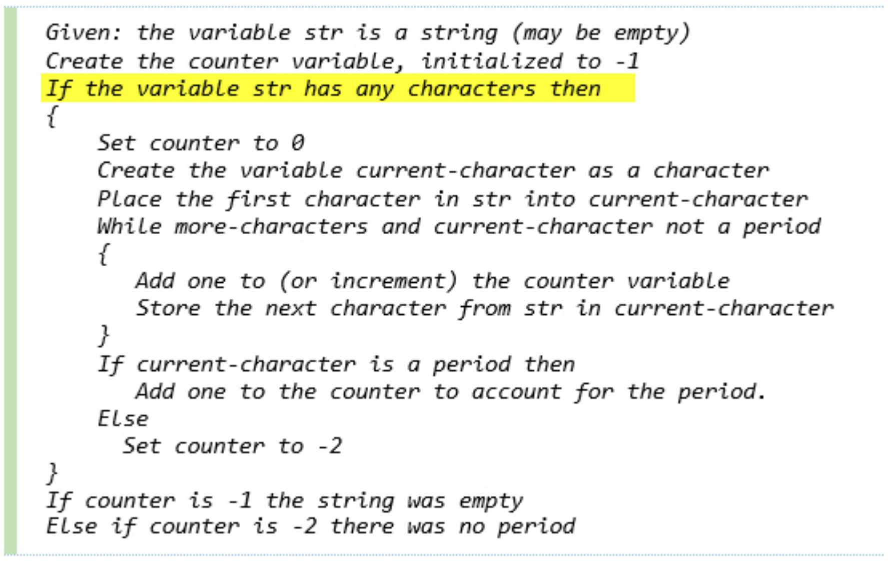
ⓘ Correct answers are hidden.

Submitted Jun 28 at 1:38am

Question 1

1 / 1 pts

The highlighted section below illustrates. If the variable str has any characters then:



```
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
```

○ a boundary condition

○ None of these

○ a necessary condition

○ a postcondition

● a loop guard

○ an intentional condition

Question 2

1 / 1 pts

Which of these is a ***flow-of-control*** statement?

☑ `while (x < 3) …`

☑ `for (auto e : s) …`

□ `int x;`

□ `x++;`

□ `int y{15};`

☑ `if (x < 3) … else …`

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 ⌄

Question 4

1 / 1 pts

In the classic *for* loop, loop control variables going from 0 to less-than n are said to employ:

○ necessary bounds

○ intentional bounds

○ None of these

○ symmetric bound

● asymmetric bounds

Question 5

1 / 1 pts

Below is the illustration from the loop building strategy. The ***highlighted lines*** represent.
While more-characters and current-character not a period:

```
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

○ goal operation

○ bounds precondition

○ advancing the loop

● loop bounds

○ loop postcondition

Question 6

1 / 1 pts

Below is the illustration from the loop building strategy. The ***highlighted lines*** represent.
If current-character is a period then:

```
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
```

● loop postcondition

○ goal operation

○ loop bounds

○ advancing the loop

○ bounds precondition

○ goal precondition

Question 7

1 / 1 pts

What prints?

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

● Does not compile

○ `Hello`

○ Undefined behavior

○ `Hell`

○ Crashes when run

Incorrect

Question 8

0 / 1 pts

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;
```

○ 2

○ 4

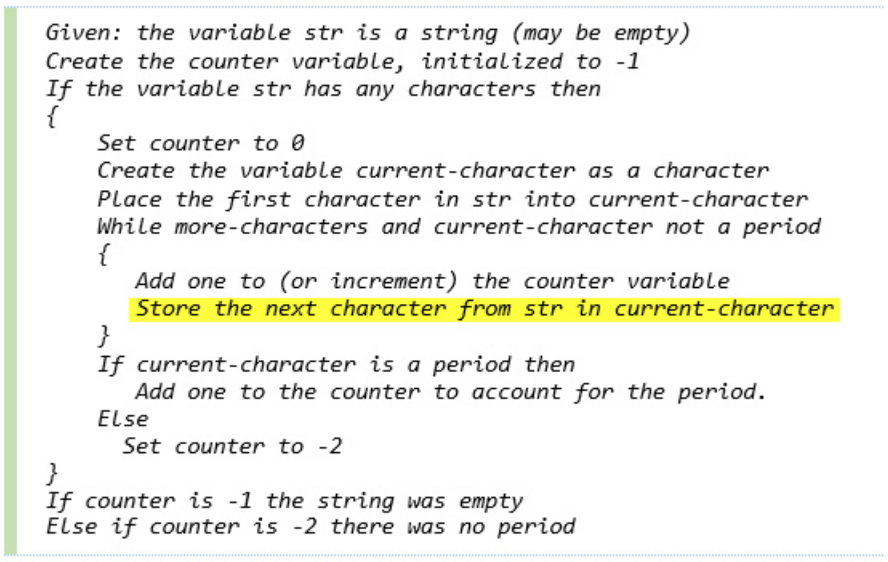
● 5

○ None of these

Question 9

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
```

○ loop bounds

○ loop postcondition

○ bounds precondition

○ goal precondition

● advancing the loop

○ goal operation

Question 10

1 / 1 pts

What kind of error is this?

```
ex1.cpp:6:12: error: no viable conversion from 'int' to 'string'
    string a = 15;
           ^   ~~
```

○ Linker error (something is missing when linking)

○ Compiler error (something is missing when compiling)

○ None of these

○ Runtime error (throws exception when running)

○ Operating system signal or trap

● Type error (wrong initialization or assignment)

○ Syntax error (mistake in grammar)

Question 11

1 / 1 pts

Match each item with the correct statement below.

Meaning of value returned from a function	@return
Begin a block of source code	@code
Information about the library	@version

Name and meaning for a parameter @param

Partial

Question 12

1 / 1 pts

In a `while` loop, (*`condition`*) is followed by a semicolon.

○ True

● False

Question 13

0.5 / 1 pts

Header guards:

□ go in every client file

□ includes the directive `#if`

☑ end with the directive `#endif`

□ go in every interface file

□ start with the directive `#ifdef`

□ go in every implementation file

☑ start with the directive `#ifndef`

□ includes the directive `#define`

Question 14

1 / 1 pts

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

● True

○ False

Question 15

1 / 1 pts

Examine this code. Which is the best prototype?

```
int age;
string name = read("Enter your name, age: ", age);
```

○ `string read(const string&, int)`

○ None of these

○ `string read(const string, int&)`

● `string read(const string&, int&)`

○ `string read(string, int);`

Incorrect

Question 16

0 / 1 pts

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`

○ `1 3 5 7 9 11 13 15 17`

● `1 3 5 7 9 11 13 17 19`

○ `1 3 5 7 9 11 13 15 17 19`

Question 17

1 / 1 pts

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&`

○ `string`

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

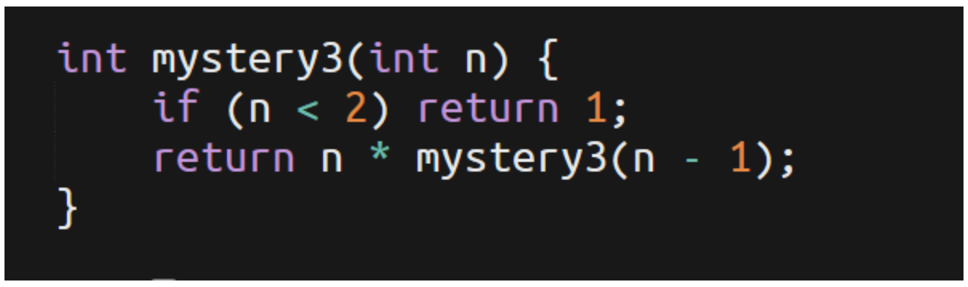
○ `const string&`

○ `const string`

Question 18

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

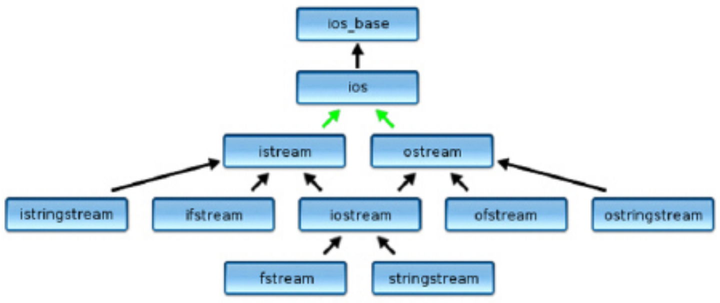
mystery3 is efficient True

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.

Question 19

1 / 1 pts

In the C++ stream hierarchy, the base class of the `stringstream` class is:



● `iostream`

○ None of these

○ `ostream`

○ `istream`

○ `fstream`

Question 20

1 / 1 pts

This loop:

```
string str;
while (in >> str)
{
    cout << str << endl;
}
```

○ illustrates raw character I/O

● illustrates token-based stream processing

○ is an endless loop

○ has a syntax error

○ illustrates line-based stream processing

Incorrect

Question 21

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

○ `3x1x750x25x`

○ `3x1x750x25x`

● `3x1x7x5x0x2x5x`

○ `3x (then cin fails)`

○ `3x1x750.25x`

Question 22

1 / 1 pts

Calling `cout.put(65)` prints the number `65` on output.

○ True

● False

Question 23

1 / 1 pts

Infinite recursion can lead to an error known as

● stack overflow

○ heap fragmentation

○ memory exception

○ heap exhaustion

Incorrect

Question 24

0 / 1 pts

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

```
char c;
cout.put(cin.get(c));
```

○ `true (or 1)`

○ Does not compile

○ `brown cow`

● `b`

Question 25

1 / 1 pts

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

```
int r(const string& s)
{
  if (s.size())
      return (s.at(0) == 'x') + r(s.substr(1));
  return 0;
}
```

● `4`

○ 2

○ 6

○ 3

○ Stack overflow