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 1	30 minutes	18.33 out of 25
LATEST	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 7:49pm

Question 1

1 / 1 pts

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

○ True

● False

Incorrect

Question 2

0 / 1 pts

Which are the two major categories of loops?

☑ definite loops

□ data loops

□ sentinel loops

☑ infinite loops

□ limit loops

□ indefinite loops

Question 3

1 / 1 pts

In the classic *for* loop, which portion of code is not followed by a semicolon?

○ condition expression

○ initialization statement

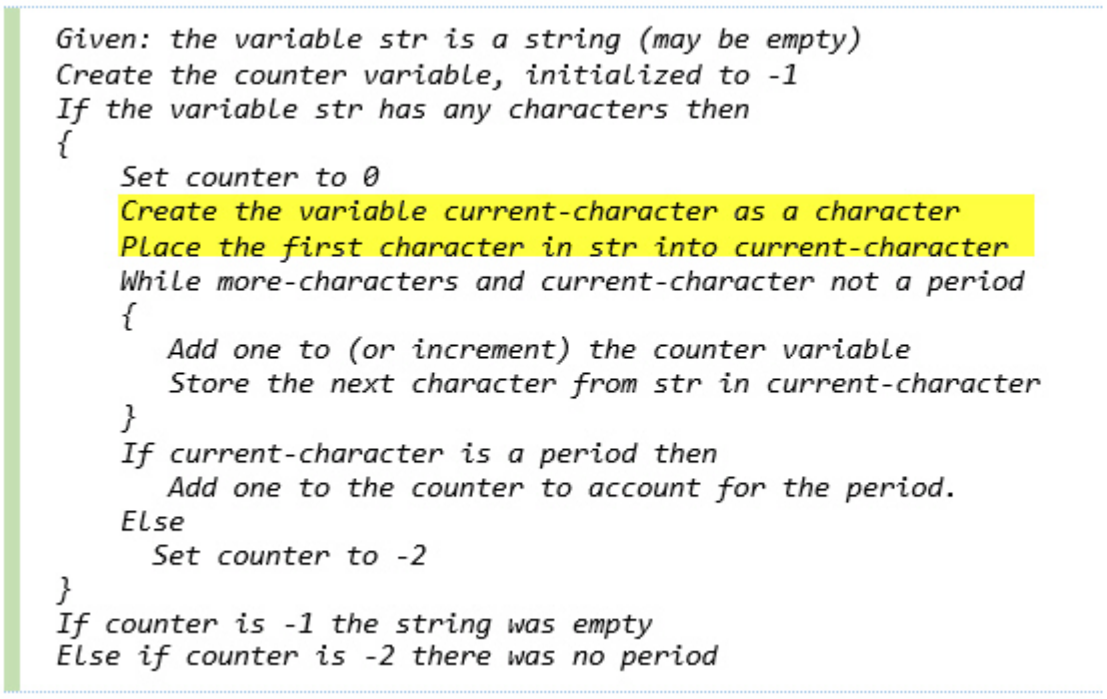
● update expression

○ None of these

Question 4

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
```

○ loop postcondition

○ goal precondition

● bounds precondition

○ loop bounds

○ goal operation

○ advancing the loop

Partial

Question 5

0.67 / 1 pts

Which of these are indefinite loops?

☑ sentinel loops

□ limit loops

☑ data loops

□ counter controlled loops

Question 6

1 / 1 pts

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

○ necessary bounds

○ None of these

● asymmetric bounds

○ symmetric bound

○ intentional bounds

Question 7

1 / 1 pts

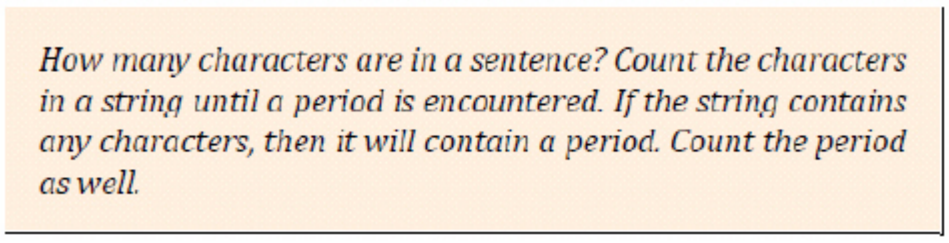
Match each item with the correct question below.

What information is produced?	goal precondition
Can my loop reach its bounds?	necessary bounds
How is the data processed?	loop operations or actions
What makes this loop quit?	loop bounds

Question 8

1 / 1 pts

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.

○ bounds

○ None of these

● goal

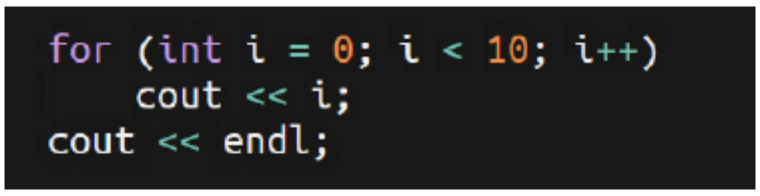
○ plan

Incorrect

Question 9

0 / 1 pts

This loop uses asymmetric bounds.



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

○ True

● False

Incorrect

Question 10

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, 2.5);
}
```

● Syntax error: ambiguous

○ B

○ C

○ A

○ Syntax error: no candidates

○ D

Question 11

1 / 1 pts

Default arguments may only be used with reference parameters.

○ True

● False

Question 12

1 / 1 pts

Which of these are ***dependencies***?

```
EXE=digit-tester
OBJS=client.o digits.o
$(EXE): $(OBJS)
    $(CXX) $(CXXFLAGS) $(OBJS) -o $(EXE)
```

□ None of these

☑ digits.o

□ $(EXE)

□ digit-tester

☑ client.o

Question 13

1 / 1 pts

What prints here?

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

○ B

○ Does not compile

○ A

○ ?

○ Prints nothing

● AB

Incorrect

Question 14

0 / 1 pts

What kind of error is this?

```
Segmentation fault
```

○ Operating system signal or trap

○ None of these

○ Type error (wrong initialization or assignment)

○ Compiler error (something is missing when compiling)

● Runtime error (throws exception when running)

○ Linker error (something is missing when linking)

○ Syntax error (mistake in grammar)

Question 15

1 / 1 pts

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

○ True

● False

Question 16

1 / 1 pts

When you **call** a function, the compiler must know:

☑ the name of the function

☑ the number of arguments to pass

□ the name of each argument

☑ the type of each argument

Question 17

1 / 1 pts

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.`

○ `// 1.`

○ None of these

○ `// 3.`

○ `// 5.`

○ `// 2.`

Question 18

1 / 1 pts

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 π. The formula states that $\frac{\pi^2}{6}$ is equal to the limit, as *n* goes to infinity, of the series $\frac{1}{1}+\frac{1}{2^2}+\frac{1}{3^2}+\ldots+\frac{1}{n^2}$. Which function below is a correct recursive implementation that approximates this infinite series?

●
```
double computePI(int number)
{
    if (number <= 1)  { return 1.0;}
    return 1.0 / (number * number) + computePI(number - 1);
}
```

○
```
double computePI(int number)
{
    if (number <= 1)  { return 1.0;}
    return (number * number) + computePI(number - 1);
}
```

○
```
double computePI(int number)
{
    if (number <= 1) { return 0.0;}
    return 1.0 / (number * number) + computePI(number - 1);
}
```

○
```
double computePI(int number)
{
    if (number <= 1)  { return 0.0;}
    return (number * number) + computePI(number - 1);
}
```

Question 19

1 / 1 pts

Calling `cout.put(c)` converts its argument, `c`, to a character.

● True

○ False

Question 20

1 / 1 pts

Assume you have a `char` variable named `ch`. How do you "unread" a character already read?

○ `cin.seek(ch);`

○ `cin.peek(ch);`

○ `cin.get(ch);`

● `cin.putback(ch);`

○ `cin.unget(ch);`

○ `None of these`

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();
        int n;
        in >> n;
        cout << n << 'x';
    }
}
```

○ `3x1x7x5x0x2x5x`

○ `3x (then cin fails)`

○ None of these

○ `3x1x750x25`

○ `3x1x750x25x`

● `3x1x750.25x`

Question 22

1 / 1 pts

What is the value of ***`mystery(12)`***?

```
int mystery(int n)
{
    if (!n) return 0;
    return 2 + mystery(n-1);
}
```

○ `36`

● `24`

○ 18

○ 12

Incorrect

Question 23

0 / 1 pts

Which command finds the word `"order"` in the file `input.txt`?

□ `find "order" < input.txt`

□ `grep "order" > input.txt`

□ `grep < input.txt "order"`

□ `None of these`

□ `cat < input.txt | grep "order"`

☑ `grep "order" | input.txt`

Incorrect

Question 24

0 / 1 pts

Which of these ***are not*** process filters?

☑ compress input by turning off echo when reading blank spaces

□ search for a particular value in a stream

□ counting word transitions

□ print one sentence per line

□ translating data from one form to another

☑ copy a file

Question 25

1 / 1 pts

Which of the following is true about using recursion?

○ A recursion eventually exhausts all available memory, causing the program to terminate.

○ None of the listed options.

○ Recursion always helps you create a more efficient solution than other techniques.

● A recursive computation solves a problem by calling itself with simpler input.