

Started on	Wednesday, September 28, 2022, 5:08 PM
State	Finished
Completed on	Wednesday, September 28, 2022, 5:49 PM
Time taken	41 mins 3 secs
Grade	43.00 out of 43.00 (100%)

Question **1**
Correct
1.00 points out of 1.00

Expression `expr1 >= expr2` evaluates as *false* if _____ (choose best possible answer)

- Select one:
- ☐ `expr2` is a smaller value than `expr1`
 - ☐ `expr2` is equivalent to `expr1`
 - ☒ `expr2` is a larger value than `expr1` ✓
 - ☐ `expr2` is a smaller value than `expr1` or is equivalent to `expr1`
 - ☐ `expr2` is a larger value than `expr1` or is equivalent to `expr1`

Your answer is correct.
The correct answer is: `expr2` is a larger value than `expr1`

Question **2**
Correct
1.00 points out of 1.00

If the expression `j != k` is *true*, then _____ (choose all correct answers)

- Select one or more:
- ☒ expression `j > k` might be *true* ✓
 - ☒ expression `j < k` might be *true* ✓
 - ☐ expression `j == k` is *true*
 - ☒ expression `j == k` is *false* ✓

Your answer is correct.
The correct answers are: expression `j > k` might be *true*, expression `j < k` might be *true*, expression `j == k` is *false*

Question **3**
Correct
1.00 points out of 1.00

If expression `x <= y` is *true*, then _____ (choose best possible answer)

- Select one:
- ☐ `y <= x` is *true*
 - ☐ `x == y` is *true*
 - ☒ `x > y` is *false* ✓
 - ☐ `x >= y` is *false*

Your answer is correct.
The correct answer is: `x > y` is *false*

Question **4**

Correct

1.00 points out of 1.00

The expression `!(x > 0)` will evaluate *true* **only** if _____. (Select the best possible answer).

Select one:

- ☐ `x` is exactly zero
- ☐ `x` is a negative value
- ☐ `x` is any value
- ☐ `x` is any positive number
- ☒ `x` is either zero or a negative number ✓

Your answer is correct.

The correct answer is: `x` is either zero or a negative number

Question **5**

Correct

1.00 points out of 1.00

The short-circuit evaluation of the operands of the `&&` operator (page 76 of text) in expression `(expr1 && expr2)` means that

Select one:

- ☐ `expr2` is first evaluated; if `expr2` evaluates *true* then `expr1` is evaluated
- ☒ `expr1` is first evaluated; if `expr1` evaluates *true* then `expr2` is evaluated ✓
- ☐ `expr2` is first evaluated; if `expr2` evaluates *false* then `expr1` is evaluated
- ☐ `expr1` is first evaluated; if `expr1` evaluates *false* then `expr2` is evaluated
- ☐ both `expr1` and `expr2` are evaluated

Your answer is correct.

The correct answer is: `expr1` is first evaluated; if `expr1` evaluates *true* then `expr2` is evaluated

Question **6**

Correct

1.00 points out of 1.00

The short-circuit evaluation of the operands of the `||` operator (page 76 of text) in expression `(expr1 || expr2)` means that

Select one:

- ☐ both `expr1` and `expr2` are evaluated
- ☒ `expr1` is first evaluated; if `expr1` evaluates *false* then `expr2` is evaluated ✓
- ☐ `expr2` is first evaluated; if `expr2` evaluates *false* then `expr1` is evaluated
- ☐ `expr2` is first evaluated; if `expr2` evaluates *true* then `expr1` is evaluated
- ☐ `expr1` is first evaluated; if `expr1` evaluates *true* then `expr2` is evaluated

Your answer is correct.

The correct answer is: `expr1` is first evaluated; if `expr1` evaluates *false* then `expr2` is evaluated

Question **7**

Correct

2.00 points out of 2.00

Often the expression in an `if` statement will test whether a variable falls *within* a range of values. To test whether `i` \in `[0, n)`, the expression in the `if` statement must be

Select one:

- ☐ `0 <= i || i < n`
- ☐ `0 <= i < n`
- ☐ `0 >= i || i >= n`
- ☐ `0 > i && i <= n`
- ☒ `0 <= i && i < n` ✓

Your answer is correct.

The correct answer is: `0 <= i && i < n`

Question **8**

Correct

2.00 points out of 2.00

Often the expression in an `if` statement will test whether a variable falls *outside* a range of values. To test whether `i` \notin `[0, n)`, the expression in the `if` statement must be

Select one:

- ☐ `i <= 0 && i > n`
- ☐ `0 <= i && i < n`
- ☒ `i < 0 || i >= n` ✓
- ☐ `0 >= i && i > n`
- ☐ `i <= 0 || i > n`

Your answer is correct.

The correct answer is: `i < 0 || i >= n`

Question **9**

Correct

1.00 points out of 1.00

In a conditional statement, the `else` clause executes _____.

Select one:

- ☐ always
- ☐ never
- ☐ when the tested condition is *true*
- ☒ when the tested condition is *false* ✓

Your answer is correct.

The correct answer is: when the tested condition is *false*

Question **10**

Correct

2.00 points out of 2.00

Many compilers may not generate a warning if the assignment (=) operator is mistakenly used instead of the equality operator (==). That is, even though you mistakenly write

```
if (x=10)
```

rather than

```
if (x==10)
```

the compiler may not generate a warning. Rewrite expression `x==10` so that the compiler will always generate an error when the assignment operator is mistakenly used instead of the equality operator? Write only the expression without using any white-space or brackets (don't write an `if` statement, instead just provide the expression). Hint: Check Chapter 5 for the answer.

Answer: 

The correct answer is: 10==x

Question **11**

Correct

1.00 points out of 1.00

Walk through the following code fragment and write the **exact** output printed to standard output. Assume variable `room_area` is of type `int` and is initialized with value 3000 while variable `painting_cost` is of type `double`.

```
0 < room_area && room_area <= 5000
? (painting_cost=40.0)
: (painting_cost = 40.0 + (room_area - 5000)*0.01);
printf("%.2f", painting_cost);
```

Answer: 

The correct answer is: 40.00

Question **12**

Correct

1.00 points out of 1.00

Walk through the following code fragment and write the exact output printed to standard output. Assume variable `room_area` is of type `int` and is initialized with value 6000 while variable `painting_cost` is of type `double`.

```
0 < room_area && room_area <= 5000
? (painting_cost=40.0)
: (painting_cost = 40.0 + (room_area - 5000)*0.01);
printf("%.2f", painting_cost);
```

Answer: 

The correct answer is: 50.00

Question 13

Correct

1.00 points out of 1.00

Walk through the following code fragment and write the **exact** output printed to standard output. Assume variable `ch` is of type `char` and is initialized with value `'u'`. Further assume that the character encoding scheme ensures that Latin characters `'a'` through `'z'` are encoded as an ordered increasing sequence of integer values. That is, if character `'a'` is encoded with integer value `100` then character `'b'` is encoded with integer value `101`, and so on. Similarly, Latin characters `'A'` through `'Z'` are also encoded as an ordered increasing sequence of integer values. The specific integer values themselves should be of no concern to a programmer.

```
if (ch >= 'A' && ch <= 'Z')
    ch += 'a' - 'A';
printf("%c", ch);
```

Answer: 

The correct answer is: u

Question 14

Correct

1.00 points out of 1.00

Often the expression in an `if` statement tests whether a variable is equivalent to a value. Write C's equality operator.

Answer: 

The correct answer is: ==

Question 15

Correct

1.00 points out of 1.00

Walk through the following code fragment and write the **exact** output printed to standard output.

```
if (60<=12*5)
    printf("Hello ");
printf("There");
```

Answer: 

The correct answer is: Hello There

Question 16

Correct

1.00 points out of 1.00

Walk through the code fragment and write the **exact** output printed to standard output.

```
if (7 <= 7)
    printf("%d", 6-9*2/6);
```

Answer: 

The correct answer is: 3

Question **17**

Correct

1.00 points out of 1.00

Walk through the following code fragment and write the ***exact*** output printed to standard output. Assume **x** is a variable of type `int` that is initialized to value 0.

```
if (x += 1)
    printf("true");
else
    printf("false");
```

Answer: ✓

The correct answer is: true

Question **18**

Correct

1.00 points out of 1.00

Often the expression in an `if` statement tests whether a variable is not equivalent to a value. Write C's inequality operator.

Answer: ✓

The correct answer is: !=

Question **19**

Correct

1.00 points out of 1.00

Walk through the following code fragment and write the exact output printed to standard output. Assume all variables are defined as type `int` with variables **x**, **y**, and **z** initialized to values 10, 15, and 20, respectively.

```
w = x != 5 && y != z;
printf("%d", w);
```

Answer: ✓

The correct answer is: 1

Question **20**

Correct

1.00 points out of 1.00

Walk through the following code fragment and write the exact output printed to standard output. Assume all variables are defined as type `int` with variables **x**, **y**, and **z** initialized to values 10, 15, and 20, respectively.

```
w = x <= y - 2 && y >= z || z - 2 != 20;
printf("%d", w);
```

Answer: ✓

The correct answer is: 1

Question **21**

Correct

1.00 points out of 1.00

Walk through the following code fragment and write the **exact** output printed to standard output. Assume that the character encoding scheme ensures that Latin characters 'a' through 'z' are encoded as an ordered increasing sequence of integer values. That is, if character 'a' is encoded with integer value 100 then character 'b' is encoded with integer value 101, and so on. Similarly, Latin characters 'A' through 'Z' are also encoded as an ordered increasing sequence of integer values. The specific integer values themselves should be of no concern to a programmer.

```
if ('a'>'b' || 'B'>'A')
    printf("#");
else
    printf("##");
```

Answer: #



The correct answer is: #

Question **22**

Correct

1.00 points out of 1.00

Walk through the following code fragment and write the **exact** output printed to standard output. Assume all variables are defined as type `int` with variables `x` and `y` initialized to values 10 and 15, respectively.

```
z = x<=5 || y<15;
printf("%d", z);
```

Answer: 0



The correct answer is: 0

Question **23**

Correct

1.00 points out of 1.00

Walk through the following code fragment and write the exact output printed to standard output. Assume all variables are defined as type `int` with variables `x`, `y`, and `z` initialized to values 10, 15, and 20, respectively.

```
w = x >= z || x + y >= z;
printf("%d", w);
```

Answer: 1



The correct answer is: 1

Question 24

Correct

5.00 points out of 5.00

Consider the (hypothetical) policy used by the Singapore Traffic Police (STP) to compute speeding fines: every ticket is \$90 plus a tiered fine determined by the offender's speed above the speed limit.

speed above maximum speed limit	fine
exceeding speed limit up to 5 mph (miles per hour)	\$65
exceeding speed limit up to 10 mph	\$150
exceeding speed limit up to 15 mph	\$230
more than 15 mph	\$350 plus \$20.50 per mile over 15 mph

Write a program to implement this speeding policy. Your program should prompt the user for both the posted speed limit and user's speed when caught in STP's speed trap. Write the speeding fine computed by your program when the posted maximum speed limit is 45 mph and the offender is traveling at 84.5 mph.

NOTE: Make sure to write this program on your own. You'll need the experience - similar questions may show up in the midterm and final tests.

Answer: 942.25

✓

The correct answer is: 942.25

Question 25

Correct

5.00 points out of 5.00

Consider the (hypothetical) policy used by the Singapore Traffic Police to compute speeding fines: every ticket is \$110 plus \$5 for each mph (miles per hour) over the speed limit, plus a penalty of \$250 for any speed over 80 mph. Write a program to implement this speeding policy. Your program should prompt the user for both the posted speed limit and user's speed when caught in STP's speed trap. Write the speeding fine computed by your program when the posted maximum speed limit is 45 mph and the offender is traveling at 84.5 mph.

NOTE: Make sure to write this program on your own. You'll need the experience - similar questions may show up in the midterm and final tests.

Answer: 557.5

✓

The correct answer is: 557.50

Question 26

Correct

1.00 points out of 1.00

The expression in the `if` statement:

```
1 if (score = 70.0)
2   grade = 'P';
3
```

always evaluates *true*.

Select one:

- ☒ True ✓
- ☐ False

The correct answer is 'True'.

Question **27**

Correct

1.00 points out of 1.00

The expression in the `if` statement:

```
1  if (score = 0.0)
2    grade = 'F';
3
```

always evaluates *true*.

Select one:

- ☐ True
- ☒ False ✓

The correct answer is 'False'.

Question **28**

Correct

1.00 points out of 1.00

Often the expression in an `if` statement tests whether a variable is equivalent to a value. However, instead of writing `if (i == 10)`, we mistakenly write `if (i = 10)`. If `i` is defined as an `int` variable, will the incorrectly written `if` statement be flagged as an error by the compiler?

Select one:

- ☐ True
- ☒ False ✓

The correct answer is 'False'.

Question **29**

Correct

1.00 points out of 1.00

Assume variable `x` is of type `int` and is initialized to value 5. Consider the following code fragment:

```
if (x < 5) printf("%d", x); x = 0; else printf("x is zero");
```

The output printed to standard output by the code fragment is:

x is zero

Select one:

- ☐ True
- ☒ False ✓

The correct answer is 'False'.

Question **30**

Correct

1.00 points out of 1.00

Every `if` statement must have a corresponding `else` clause.

Select one:

- ☐ True
- ☒ False ✓

The correct answer is 'False'.

Question **31**

Correct

1.00 points out of 1.00

Assuming `ch` is defined as a `char` variable, the expression

```
ch >= 'A' && ch <= 'Z'
```

evaluates *false* if either `ch < 'A'` or `ch > 'Z'`

Select one:

- ☒ True ✓
- ☐ False

The correct answer is 'True'.

Question **32**

Correct

1.00 points out of 1.00

A relational and equality expression contains relational and equality operators (`<`, `<=`, `>`, `>=`, `==`, `!=`). The results obtained by evaluating such relational or equality expressions cannot be assigned to an `int` variable.

Select one:

- ☐ True
- ☒ False ✓

The correct answer is 'False'.

◀ Quiz 4: Arithmetic, Relational, and Logical Operators

Jump to...

Lab 4: Formatted Console I/O ▶