

Quiz 2 Solutions

Q1 (1 pt): If `grade` has the value of 60, what will the following code display?

```
if ( grade >= 60 )
    cout << "Passed";
```

- nothing.
- 60
- Passed
- `cout << "Passed";`

ANS: c. **Passed.**

Q2 (2 pt): What is wrong with the following `while` loop?

```
while ( sum <= 1000 )
    sum = sum - 30;
```

- The parentheses should be braces.
- Braces are required around `sum = sum - 30;`.
- There should be a semicolon after `while (sum <= 1000)`.
- `sum = sum - 30` should be `sum = sum + 30` or else the loop may never end.

ANS: d. **sum = sum - 30 should be sum = sum + 30 or else the loop may never end.**

Q3 (1 pt): An uninitialized local variable contains:

- The value last stored in the memory location reserved for that variable.
- No value.
- A value of zero.
- A randomly assigned value.

ANS: a. **The value last stored in the memory location reserved for that variable.**

Q4 (2 pt): What is the final value of `x` after performing the following operations?

```
int x = 21;
double y = 6;
double z = 14;
y = x / z;
x = 5.5 * y;
```

- 8.25.
- 5.5.
- 5.
- 8.

ANS: d. **8.**

Q5 (1 pt): Which of the following `for` headers is *not* valid?

- `for (int i = 0; i < 10; i++)`
- `int i = 0;`
`for (; i < 10; i++)`
- `for (int i = 0; int j = 5; ; i++)`
- All of the above.

ANS c. **for (int i = 0; int j = 5; ; i++).**

Q6 (1 pt): `switch` can be used to test:

- `int` constants.
- `float` constants.
- `string` constants.
- all types of constants.

ANS: a. **int constants.**

EECS 211: FUNDAMENTALS OF COMPUTER PROGRAMMING II

Q7 (1 pt): In C++, the condition `(4 > y > 1)`:

- Evaluates correctly and could be replaced by `(4 > y && y > 1)`.
- Does not evaluate correctly and should be replaced by `(4 > y && y > 1)`.
- Evaluates correctly and could not be replaced by `(4 > y && y > 1)`.
- Does not evaluate correctly and should not be replaced by `(4 > y && y > 1)`.

ANS: b. Does not evaluate correctly and should be replaced by `(4 > y && y > 1)`.

Q8 (1 pt): The OR `(| |)` operator:

- Has higher precedence than the AND `(&&)` operator.
- Stops evaluation upon finding one condition to be `true`.
- Associates from right to left.
- Is a ternary operator.

ANS: b. Stops evaluation upon finding one condition to be `true`.

Q9 (1 pt): The function prototype

```
double mysqrt( int x );
```

- Declares a function called `mysqrt` which takes an integer as an argument and returns a `double`.
- Defines a function called `double` which calculates square roots.
- Defines a function called `mysqrt` which takes an argument of type `x` and returns a `double`.
- Declares a function called `mysqrt` which takes a `double` as an argument and returns an integer.

ANS: a. Declares a function called `mysqrt` which takes an integer as an argument and returns a `double`.

Q10 (1 pt): Which of the following is *not* true of `static` local variables?

- They are accessible outside of the function in which they are defined.
- They retain their values when the function in which they are defined terminates.
- They are initialized to zero if not explicitly initialized by the programmer.
- They can be of type `int`.

ANS: a. They are accessible outside of the function in which they are defined.

Q11 (1 pt): What happens when two blocks, one nested inside of the other, both declare variables with the same identifier? (Assume that the outer block declares its variable before the opening left-brace of the inner block.)

- A syntax error occurs.
- The “outer” variable is hidden while the “inner” variable is in scope.
- The “outer” variable is irretrievably lost when the “inner” variable is declared.
- The “inner” declaration is ignored and the “outer” variable has scope even inside the inner block.

ANS: b. The “outer” variable is hidden while the “inner” variable is in scope.

Q12 (1 pt): The `inline` keyword:

- Increases function-call overhead.
- Can reduce a function’s execution time but increase program size.
- Can decrease program size but increase the function’s execution time.
- Should be used with all frequently used functions.

ANS: b. Can reduce a function’s execution time but increase program size.

Q13 (2 pt): When an argument is passed-by-value, changes in the called function _____ affect the original variable’s value; when an argument is passed call-by-reference, changes in the called function _____ affect the original variable’s value.

- Do not, do.
- Do not, do not.
- Do, do.
- Do, do not.

ANS: a. Do not, do.

EECS 211: FUNDAMENTALS OF COMPUTER PROGRAMMING II

Q14 (1 pt): An array is *not*:

- a. A consecutive group of memory locations.
- b. Subscripted by integers.
- c. Declared using braces, `[]`.
- d. Made up of different data types.

ANS d. Made up of different data types.

Q15 (1 pt): Which statement would be used to declare a 10-element integer array `c`?

- a. `array c = int[10];`
- b. `c = int[10];`
- c. `int array c[10];`
- d. `int c[10];`

ANS d. `int c[10];`

Q16 (1 pt): Referencing elements outside the array bounds:

- a. Can result in changes to the value of an unrelated variable.
- b. Is impossible because C++ checks to make sure it does not happen.
- c. Is a syntax error.
- d. Enlarges the size of the array.

ANS: a. Can result in changes to the value of an unrelated variable.

Q17 (1 pt): Unless otherwise specified, entire arrays are passed _____ and individual array elements are passed _____.

- a. By value, by reference.
- b. By reference, by value.
- c. By value, by value.
- d. By reference, by reference.

ANS: b. By reference, by value.