Quiz >

Review answers

Start date:	7 minutes ago
Complete date:	1 minute ago

C++:

```
1  int* p1, p2;
```

◯ Only *p1* is a variable containing a pointer to an *int*. Variable *p2* contains an *int* directly.
◯ This code does not compile.
⊘ Both *p1* and *p2* are variables containing a pointer to an *int*.
◯ Only *p2* is a variable containing a pointer to an *int*. Variable *p1* contains an *int* directly.

Question 2: Which two statements are true about files?

☑ File IO can be done with the file versions of IO functions like *fprintf()* and *fscanf()* that accept a *FILE** as extra argument.
☑ When finished with a file, you need to close it with the *fclose()* function.
☐ Before you can read or write a file, you need to open it first using the *open()* function.
☐ The standard input/output functions like *printf()* and *scanf()* can be used for file IO by just passing a *FILE** as first argument.

Question 3: Which statement is true about the following program?

C++:

```
1  #define sqr(x) (x * x)
2
3  int main()
4  {
5    printf("%.2f, ", sqr(10.0));
6    printf("%.2f\n", sqr(10.0-5.0));
```

```
}
```

✓ The output of this program is: 100.00, -45.00
○ This program does not compile because the define is missing round brackets around the x.
○ The output of this program is: 100.00, 25.00
○ This program does not compile because the define is missing the semi-colon (;) at the end of the line.

Question 4: How do you retrieve the length of an array variable defined as: int arr[10]?

○ Using the *arr.length* data member.
✓ C does not store the length of an array, so it can't be retrieved.
○ Using the *array_length(arr)* function.
○ Using the *sizeof(arr)* operator.

Question 5: Which of the following typedef statements below defines a pointer to an array of 10 integers?

✗ typedef int *ArrayPointer[];
○ typedef int (*ArrayPointer[]);
○ typedef int*[] ArrayPointer;
○ typedef int (*ArrayPointer)[];

Question 6: Which of the option(s) below are correct forms to access the third element of the array in the flowing code?

C++:

```
1 int array[10]; int i;
```

☐ i=*array+2;
☑ i=array[2];
☑ i=(array+2)[0];
☑ i=*(array+2);

Question 7: Which of the function declarations below is the fastest way to pass a struct type called *MyStruct* to a function?

○ void F(MyStruct s);

	✓ void F(struct MyStruct* s); ○ void F(struct MyStruct s); ○ void F(MyStruct* s);
Question 8:	Which two statements are true about typedefs? ☐ Typedefs are a pre-processor construct to replace an identifier with the specified code. ☐ Typedefs can be used to define constant values. ☑ Typedefs can be used to make it simpler to port your code to other platforms that have different implementations of certain types. ☑ Typedefs are a compiler construct to give another (simpler) name to a (complex) type definition.
Question 9:	Which statement is false about the pre-processor? ○ The pre-processor adapts your source file before it is processed by the compiler. ○ The pre-processor can replace an identifier with other text. ○ The pre-processor can insert another file in a source file. ✓ Pre-processor instructions start with '#' and end with a ';'
Question 10:	Which statement is false about console input? ○ The standard input is by default the console (keyboard) but it can be redirected to read from a file. ○ The function *scanf()* can convert strings read from the standard input into the correct type. For that it needs format specifiers similar as those for the *printf()* function. ✗ Console input is buffered and input will only send to the program after the user pressed <enter>. Thus the *getchar()* function reads a single character from the standard input only after the user pressed <enter>. ○ When a single *scanf()* statement does not read all data from a line entered on the console, the rest of the line will be discarded.
Score:	7 (70.00%)
Pass/Fail:	Passed (in previous attempts)

Quiz ›