# Q-18

**Due** No due date **Points** 15 **Questions** 15 **Time Limit** 30 Minutes **Allowed Attempts** Unlimited

Take the Quiz Again

## Attempt History

| | Attempt | Time | Score |
|---|---|---|---|
| KEPT | Attempt 6 | 11 minutes | 15 out of 15 |
| LATEST | Attempt 6 | 11 minutes | 15 out of 15 |
| | Attempt 5 | 13 minutes | 15 out of 15 |
| | Attempt 4 | 14 minutes | 15 out of 15 |
| | Attempt 3 | 12 minutes | 14 out of 15 |
| | Attempt 2 | 16 minutes | 15 out of 15 |
| | Attempt 1 | 21 minutes | 14 out of 15 |

ⓘ Correct answers are hidden.

Submitted Jul 22 at 11:50pm

### Question 1

1 / 1 pts

What is printed when you run this code?

```
int *n{nullptr};
cout << &n << endl;
```

- The word "nullptr"
- No output; compiler error.
- **The address value where *n* is stored**
- The address value 0
- No compilation errors, but undefined behavior

### Question 2

1 / 1 pts

What is a common pointer error?

- Assigning a new value to a pointer
- Using indirection on a pointer
- **Using a pointer without first initializing it**
- Dereferencing a pointer
- Setting a pointer value to nullptr

### Question 3

1 / 1 pts

Assume that ***ppi*** correctly points to ***pi***. Which line prints the ***size*** (in bytes) of ***pi***?

```
int main()
{
    double pi = 3.14159;
    double *ppi;
    // code goes here
    // code goes here
}
```

- cout << sizeof(ppi);
- cout << sizeof(*pi);
- cout << sizeof(&ppi);
- **cout << sizeof(*ppi);**
- None of these

### Question 4

1 / 1 pts

What is true about this code?

```
int n{500};
int *p = &n;
```

- &p represents the indirect value of n

- p stores the same value as n
- *p is the value of n
- &n is the indirect value of p
- &p is the direct or explicit value of n

## Question 5

1 / 1 pts

What is printed when you run this code?

```
int num = 0;
int *ptr = &num;
num = 5;
*ptr += 5;
cout << num << " " << *ptr << endl;
```

- 5 10
- 5 5
- 10 10
- Undefined; none of these
- 10 5

## Question 6

1 / 1 pts

What is printed when you run this code?

```
int x(100);
cout << &x << endl;
```

- The memory location where x is stored
- The value of x (100)
- None of these
- The value stored at address 100

## Question 7

1 / 1 pts

All of these are legal C++ statements; which of them uses the C++ ***dereferencing operator***?

```
int a = 3, b = 4;
```

- None of these use the dereferencing operator.
- int y = a * b;
- int x = *p;
- z *= a;
- int *p = &b;

## Question 8

1 / 1 pts

What is printed when you run this code?

```
int n{};
int *p = &n;
*p = 10;
n = 20;
cout << *p << endl;
```

- None of these
- 20
- The address of n
- 10
- 0

## Question 9

1 / 1 pts

What is the term used to describe a variable which stores a memory address?

- (●) `pointer`
- ( ) None of these
- ( ) `rvalue`
- ( ) `lvalue`
- ( ) `reference`

## Question 10

1 / 1 pts

What is true about this code?

```
int * choice;
```

- ( ) `choice` currently points to an integer
- ( ) `choice` can point to any kind of object
- ( ) `choice` currently contains an integer
- (●) `choice` contains an undefined address
- ( ) Syntax error; should be `int choice*;`

## Question 11

1 / 1 pts

Assume that ***ppi*** correctly points to ***pi***. Which line prints the address of ***ppi***?

```
int main()
{
    double pi = 3.14159;
    double *ppi;
    // code goes here
    // code goes here
}
```

- ( ) `cout << &pi;`
- (●) `cout << &ppi;`
- ( ) `cout << ppi;`
- ( ) `cout << *ppi;`
- ( ) None of these

## Question 12

1 / 1 pts

Assume that ***p1*** is a pointer to an integer and ***p2*** is a pointer to a second integer. Both integers appear inside a large contiguous sequence in memory, with ***p2*** storing a larger address. How many total integers are there in the slice between ***p1*** and ***p2***?

- ( ) `p2 - p1 - 1;`
- ( ) None of these
- (●) `p2 - p1;`
- ( ) `p1 - p2;`
- ( ) `p1 - p2 + 1;`

## Question 13

1 / 1 pts

Examine the following code. What is stored in ***c*** after it runs.

```
int f(int * p, int x)
{
    *p = x * 2;
    return x / 2;
}
. . .
int a = 3, b, c;
c = f(&b, a);
```

- ( ) 3
- ( ) 6
- ( ) Does not compile
- ( ) 2
- (●) 1

## Question 14

1 / 1 pts

Here is a fragment of pseudocode for the ***negative()*** function from your homework. What statement represents the underlined portion of code?

```
Let p point to beginning of the image
Let end be pixel one past the end of the image
While p != end
    Invert the red component
    Move p to next component
```

- *p = p + 1;
- p++;
- *p++;
- None of these
- &p++;

## Question 15

1 / 1 pts

Examine the following code. What is stored in ***b*** after it runs.

```
int f(int * p, int x)
{
    *p = x * 2;
    return x / 2;
}
. . .
int a = 3, b, c;
c = f(&b, a);
```

- Does not compile
- 1
- 3
- 2
- 6