

# C++ Pointer Deep Dive with MCQs @

### MCQ 01: What Will Be The Output?

```
#include <bits/stdc++.h>
#include <iostream>
using namespace std;
int main() {
 float f = 10.5;
  float p = 2.5;
  float *ptr = &f;
  (*ptr)++;
  *ptr = p;
  cout << *ptr << " " << f << " " << p << endl;</pre>
```

#### Explanation:

```
1. float f = 10.5; \rightarrow f = 10.5
2. float p = 2.5; \rightarrow p = 2.5
3. float *ptr = &f; → pointer ptr points to f
4. (*ptr)++; \rightarrow value at ptr becomes 11.5
5. *ptr = p; \rightarrow value at ptr is now 2.5 \rightarrow so f = 2.5
```

### **⊘** Output:

```
2.5 2.5 2.5
```

### MCQ 02: More Pointer Modification

```
int a = 7;
int b = 17;
int *c = \&b;
*c = 7;
cout << a << " " << b << endl;</pre>
```

### Explanation:

```
1. a = 7, b = 17
2. *c = 7 changes value at address of b \rightarrow now b = 7
3. a remains unchanged
```

#### **⊘** Output:

```
7 7
```

## MCQ 03: Dangerous Code with Wild Pointer

```
int *ptr = 0;  // X Wild pointer: points to NULL
int a = 10;
*ptr = a;  // X Segmentation fault (writing to NULL)
cout << *ptr << endl;</pre>
```

### **X** Explanation:

- int \*ptr = 0; → ptr is a null pointer
- \*ptr = a; → X You are dereferencing a null pointer → crash / segfault

#### **⚠** Output:

```
Segmentation Fault / Runtime Crash
```

#### Fix It:

```
int a = 10;
int *ptr = &a; // ✓ Properly initialized
*ptr = a;
cout << *ptr << endl;</pre>
```

## MCQ 04: Pointer with Char Data Type

```
int b = 10;
int *a = &b;

char ch = 'a';
char *ptr = &ch;
ch++;
cout << *ptr << endl;</pre>
```

### Explanation:

```
    char ch = 'a';
    char *ptr = &ch; → pointer to character
    ch++ → now ch = 'b'
    *ptr still points to ch → prints 'b'
```

### **⊘** Output:

```
b
```

## **(2)** Concepts Covered

♦ 1. Pointer Reassignment and Value Change

- You can change what a pointer points to, or what it holds.
- ♦ 2. Pointer Dereferencing

```
int *c = &b;
*c = 7;
```

- \*c accesses and modifies the **value** of b because c points to b.
- ♦ 3. **X** Dangling/Wild/Null Pointer Use

```
int *ptr = 0;
*ptr = a;
```

- **⊘** You are writing to memory address **⊘** (null).
- 🌣 Leads to Segmentation Fault
- ♦ 4. Character Pointer & Increment

```
char ch = 'a';
ch++; // ch = 'b'
```

→ Character arithmetic works as ASCII manipulation ('a' + 1 = 'b').

### 

<b>≚</b> Mistake	Why It's Bad
Using *ptr = value; when ptr is null	<b>≯</b> Segfault
Declaring multiple pointers with same name	X Compiler Error
Returning local variable address	👌 Dangling Pointer
Forgetting to initialize pointer	Wild Pointer

### **☑** Best Practices with Pointers

- Always initialize pointers (nullptr if not pointing yet)
- Never dereference uninitialized or null pointers
- After delete, set pointer to nullptr to avoid dangling references
- Prefer smart pointers in C++11+ (unique\_ptr, shared\_ptr)

## Real-World Analogy

Pointer Concept	Analogy
Pointer to variable	Address to a house 🏠
Dereferencing *ptr	Entering the house to see what's inside
Wild Pointer	A key 🤌 with no door to fit
Dangling Pointer	A demolished house, but you still have the old address 🎋

# Final Output Summary of All Working Segments

```
2.5 2.5 2.5
7 7
Segmentation Fault (or crash)
b
```

### Pointer Tip Box

- \*ptr = value; → assign value to pointed address
- ptr = &var; → point to variable
- (\*ptr)++ → increment value at address

• ptr++ → move pointer to next memory (careful!)



#### **Darshan Vasani**

**Parity** Full-Stack Developer | Software Engineer | Mentor

Connect with me!

Portfolio: dpvasani56.vercel.app
GitHub: github.com/dpvasani

LinkedIn: linkedin.com/in/dpvasani56

**Linktree:** linktr.ee/dpvasani56

Credly: credly.com/users/dpvasani57
 Twitter: x.com/vasanidarshan56
 Topmate: topmate.io/dpvasani56