

## 1. Pointer Declaration and Assignment

```
#include <iostream>

int main() {

    int x = 10;

    int *ptr = &x;

    std::cout << "Value of x (direct): " << x << std::endl;

    std::cout << "Value of x (using ptr): " << *ptr << std::endl;

    return 0;

}
```

## 2. Pointer Dereferencing

```
#include <iostream>

int main() {

    float f = 5.25;

    float *fptr = &f;

    std::cout << "Value of f (direct): " << f << std::endl;

    std::cout << "Value of f (using fptr): " << *fptr << std::endl;

    std::cout << "Address of f: " << &f << std::endl;

    return 0;

}
```

## 3. Pointer to Pointer

```
#include <iostream>

int main() {

    int num = 30;

    int *p = &num;

    int **pp = &p;

    std::cout << "Value of num (direct): " << num << std::endl;

    std::cout << "Value of num (using pp): " << **pp << std::endl;

    std::cout << "Address stored in p: " << p << std::endl;

    std::cout << "Address stored in pp: " << pp << std::endl;

    return 0;

}
```

#### **4. Changing Variable Value Through a Pointer**

```
#include <iostream>

int main() {

    int val = 50;

    int *pval = &val;

    std::cout << "Initial value of val: " << val << std::endl;

    *pval = 20;

    std::cout << "New value of val (changed through pval): " << val << std::endl;

    return 0;

}
```

```
}
```

## 5. Null Pointer

```
#include <iostream>
```

```
int main() {
```

```
    int *p = nullptr;
```

```
    if (p == nullptr) {
```

```
        std::cout << "p is a null pointer." << std::endl;
```

```
    }
```

```
    int x = 15;
```

```
    p = &x;
```

```
    std::cout << "Address of x stored in p: " << p << std::endl;
```

```
    return 0;
```

```
}
```

## 6. Array and Pointer Basics

```
#include <iostream>
```

```
int main() {
```

```
    int arr[3] = {1, 2, 3};
```

```
    int *p = arr; // Points to the first element of the array
```

```
    std::cout << "First element of arr (using p): " << *p << std::endl;
```

```
return 0;
```

```
}
```