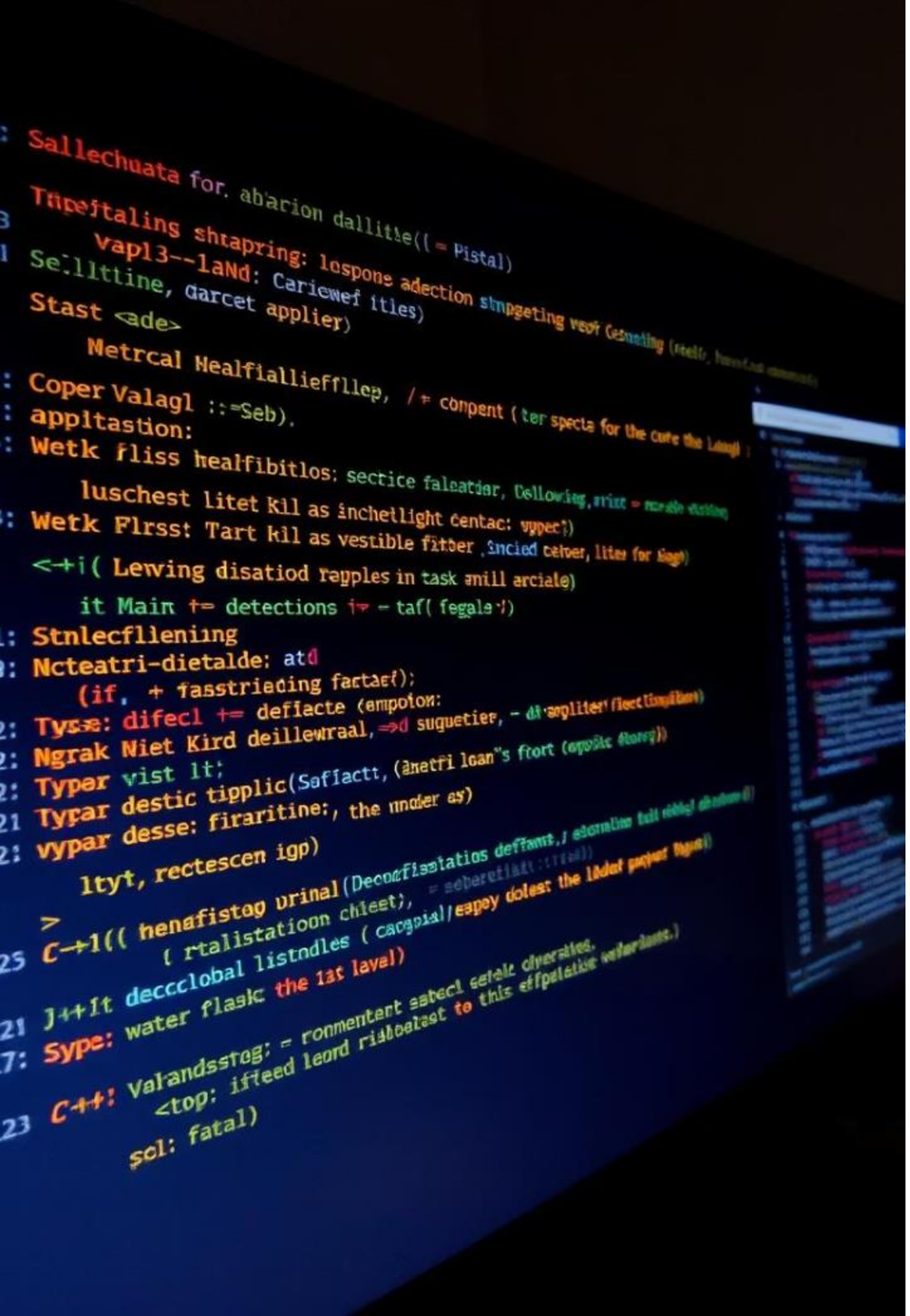

Functions and Structures in C++

Lecture 8

Asst. Lect. Ali Al-khawaja

Introduction

Functions

Q: Why do we use functions?

To divide the program into reusable and manageable parts.

Structures

Q: Why do we use structures?

To group related variables (like name, age, id).

Passing Structures to Functions

Pass Structure by Value

```
#include
using namespace std;

struct Student {
    string name;
    int age;
};

void display(Student s) {
    cout << "Name: " << s.name << ", Age: " << s.age << endl;
}

int main() {
    Student s1 = {"Ali", 20};
    display(s1); // Pass by value
    return 0;
}
```

Return Structure from Function

1

Define the Function

```
Student createStudent(string n, int a) {
    Student s;
    s.name = n;
    s.age = a;
    return s;
}
```

2

Call in Main

```
Student s2 = createStudent("Zainab", 19);
display(s2);
```

3

Use the Result

The returned structure can be used like any other structure variable.

Combining It All

1 Define the Structure

```
#include
using namespace std;

struct Book {
    string title;
    string author;
    int year;
};
```

2 Create Display Function

```
void printBook(Book b) {
    cout << "Title: " << b.title << ", Author: " << b.author << ", Year: " << b.year << endl;
}
```

3 Create Input Function

```
Book inputBook() {
    Book b;
    cout << "Enter title: ";
    getline(cin, b.title);
    cout << "Enter author: ";
    getline(cin, b.author);
    cout << "Enter year: ";
    cin >> b.year;
    cin.ignore(); // Clear input buffer
    return b;
}
```

4 Use in Main Function

```
int main() {
    Book myBook = inputBook();
    printBook(myBook);
    return 0;
}
```

Homework

- Create a structure Employee with fields: name, salary, and department.
- Write a function to read employee data.
- Write another to display it.