

Getting User Input Data

Introduction

- Getting user input is essential for interactive programs.
- In C++ and Go, different methods are used to obtain user input.

C++ Input (cin)

- In C++, you use the `cin` object from the `<iostream>` library to read user input.

Example (C++):

```
#include <iostream>
using namespace std;

int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;
    cout << "You entered: " << num << endl;
    return 0;
}
```

Go Input (fmt.Scan)

- In Go, you use the `fmt` package to read user input.

Example (Go):

```
package main

import (
    "fmt"
)

func main() {
    var num int
    fmt.Print("Enter a number: ")
    fmt.Scan(&num)
    fmt.Println("You entered:", num)
}
```

C++ Input (getline)

- For string input in C++, you can use the `getline` function.

Example (C++):

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string name;
    cout << "Enter your name: ";
    getline(cin, name);
    cout << "Hello, " << name << "!" << endl;
    return 0;
}
```

Go Input (bufio.NewReader)

- For string input in Go, you can use the `bufio` package along with `os.Stdin`.

Example (Go):

```
package main

import (
    "bufio"
    "fmt"
    "os"
)

func main() {
    reader := bufio.NewReader(os.Stdin)
    fmt.Print("Enter your name: ")
    name, _ := reader.ReadString('\n')
    fmt.Printf("Hello, %s!\n", name)
}
```

Summary

- In C++, `cin` is used for input, and `getline` is used for strings.
- In Go, `fmt.Scan` is used for input, and `bufio.NewReader` is used for strings.
- Both languages provide mechanisms to interact with the user.

Questions

Any questions on getting user input in C++ and Go?

These slides provide code examples in both C++ and Go to demonstrate how to obtain user input, both for numerical and string input. You can use these slides to compare and contrast the input mechanisms in both languages during your teaching.