Understanding Structs in Go

Custom Data Types and Object Creation

What are Structs?

- Structs are composite data types in Go that group together variables (fields) under a single name.
- They allow you to create custom data structures and objects.

Declaring a Struct in Go

- Declare a struct using the `type` keyword.

```go
type Person struct {
    FirstName string
    LastName  string
    Age       int
}
```

- `Person` is a custom struct type with fields.

Creating Instances in Go

- Create struct instances using the struct type's name and field values.

```go
alice := Person{
    FirstName: "Alice",
    LastName:  "Smith",
    Age:       30,
}
```

- Access fields using the `.` operator.

Equivalent in Python

- In Python, classes define custom data types with fields.

```python
class Person:
    def __init__(self, first_name, last_name, age):
        self.first_name = first_name
        self.last_name = last_name
        self.age = age

alice = Person("Alice", "Smith", 30)
```

Equivalent in C++

- In C++, classes define custom data types with members.

```cpp
class Person {
public:
    std::string FirstName;
    std::string LastName;
    int Age;
};

Person alice;
alice.FirstName = "Alice";
alice.LastName = "Smith";
alice.Age = 30;
```

Struct Methods in Go

- You can define methods associated with a struct in Go.

```
func (p Person) FullName() string {
    return p.FirstName + " " + p.LastName
}
```

- `(p Person)` is the receiver, allowing access to struct fields.

Struct Methods in Python

- Python class methods are similar to Go struct methods.

```python
class Person:
    def __init__(self, first_name, last_name, age):
        self.first_name = first_name
        self.last_name = last_name
        self.age = age

    def full_name(self):
        return f"{self.first_name} {self.last_name}"
```

Struct Methods in C++

- C++ uses member functions within classes for similar behavior.

```
class Person {
public:
    std::string FirstName;
    std::string LastName;
    int Age;

    std::string FullName() {
        return FirstName + " " + LastName;
    }
};
```

Struct Embedding in Go

- Go supports struct embedding for composition.

```go
type Address struct {
    Street  string
    City    string
    ZipCode string
}

type Person struct {
    FirstName string
    LastName  string
    Age       int
    Address   // Embedding Address struct
}
```

- Fields from `Address` are accessible directly on `Person`.

Struct Embedding in Python

- Python allows composition using attributes.

```python
class Address:
    def __init__(self, street, city, zipcode):
        self.street = street
        self.city = city
        self.zipcode = zipcode

class Person:
    def __init__(self, first_name, last_name, age, address):
        self.first_name = first_name
        self.last_name = last_name
        self.age = age
        self.address = address
```

Struct Embedding in C++

- C++ supports composition using objects within classes.

```
class Address {
public:
    std::string Street;
    std::string City;
    std::string ZipCode;
};

class Person {
public:
    std::string FirstName;
    std::string LastName;
    int Age;
    Address AddressObj; // Embedding Address class
};
```

Summary

- Structs in Go allow custom data types with fields.
- Methods can be associated with structs.
- Struct embedding enables composition.
- Python and C++ offer similar concepts with classes.