

## GO LANG LAB 1 - 4

PAVITHARANI G P  
2347244

Utilize the principles of Variables, Control Flow, Arrays, Slices, Maps, and Structs within a specified domain of your choice. Begin the program with a comprehensive overview detailing the scenario and the concepts being implemented. Ensure adequate comments are provided throughout the code. Evaluation of this program will be based on the following criteria:

R1: Concept Clarity/Viva: 8 Marks

R2: Correctness: 8 Marks

R3: Validations( only with if statement expected): 8 Marks

R4: Ability to Relate to Real-Time Scenario: 8 Marks

R5: Complexity: 8 Marks

CODE :

```
package main

import (
    "fmt"
)

// Item represents a photography prop item in the store.
type Item struct {
    ID        int
    Name      string
    Quantity  int
    Price     float64
}

// Store represents a photography prop store.
type Store struct {
    inventory map[int]Item // inventory maps item ID to Item
}

// NewStore creates and returns a new instance of Store.
func NewStore() *Store {
    return &Store{
        inventory: make(map[int]Item),
    }
}
```

```

// AddItem adds a new item to the store's inventory.
func (s *Store) AddItem(id int, name string, quantity int, price float64) {
    newItem := Item{
        ID:      id,
        Name:     name,
        Quantity: quantity,
        Price:    price,
    }
    s.inventory[id] = newItem
}

// SellItem attempts to sell an item from the store's inventory.
// It returns true if the item was successfully sold, and false otherwise.
func (s *Store) SellItem(id int, quantity int) bool {
    item, exists := s.inventory[id]
    if !exists {
        fmt.Println("Item not found in inventory")
        return false
    }

    if item.Quantity < quantity {
        fmt.Println("Insufficient quantity in inventory")
        return false
    }

    // Update quantity
    item.Quantity -= quantity
    s.inventory[id] = item
    fmt.Printf("Sold %d units of %s\n", quantity, item.Name)
    return true
}

func main() {
    // Create a new store
    store := NewStore()

    // Prompt user to add items
    fmt.Println("Add items to inventory:")
    for {
        var id, quantity int
        var name string
        var price float64

        fmt.Print("Enter item ID (0 to stop): ")
        fmt.Scanln(&id)
    }
}

```

```
        if id == 0 {
            break
        }

        fmt.Print("Enter item name: ")
        fmt.Scanln(&name)

        fmt.Print("Enter item quantity: ")
        fmt.Scanln(&quantity)

        fmt.Print("Enter item price: ")
        fmt.Scanln(&price)

        store.AddItem(id, name, quantity, price)

        // Ask if user wants to add more items
        var addMore string
        fmt.Print("Do you want to add more items? (yes/no): ")
        fmt.Scanln(&addMore)
        if addMore != "yes" {
            break
        }
    }

    // Display initial inventory
    fmt.Println("\nInitial Inventory:")
    for _, item := range store.inventory {
        fmt.Printf("ID: %d, Name: %s, Quantity: %d, Price: $%.2f\n", item.ID,
item.Name, item.Quantity, item.Price)
    }

    // Prompt user to sell items
    fmt.Println("\nSell items from inventory:")
    for {
        var id, quantity int

        fmt.Print("Enter item ID to sell (0 to stop): ")
        fmt.Scanln(&id)
        if id == 0 {
            break
        }

        fmt.Print("Enter quantity to sell: ")
        fmt.Scanln(&quantity)
```

```

        store.SellItem(id, quantity)
    }

    // Display updated inventory
    fmt.Println("\nUpdated Inventory:")
    for _, item := range store.inventory {
        fmt.Printf("ID: %d, Name: %s, Quantity: %d, Price: $%.2f\n", item.ID,
            item.Name, item.Quantity, item.Price)
    }
}

```

## OUTPUT :

```

gppavitharani@Its-Paviii LAB1-4 % go run lab.go
Add items to inventory:
Enter item ID (0 to stop): 1
Enter item name: Camera
Enter item quantity: 10
Enter item price: 887
Do you want to add more items? (yes/no): yes
Enter item ID (0 to stop): 3
Enter item name: Backdrop
Enter item quantity: 15
Enter item price: 234
Do you want to add more items? (yes/no): yes
Enter item ID (0 to stop): 5
Enter item name: Basket
Enter item quantity: 9
Enter item price: 100
Do you want to add more items? (yes/no): no

Initial Inventory:
ID: 1, Name: Camera, Quantity: 10, Price: $887.00
ID: 3, Name: Backdrop, Quantity: 15, Price: $234.00
ID: 5, Name: Basket, Quantity: 9, Price: $100.00

Sell items from inventory:
Enter item ID to sell (0 to stop): 3
Enter quantity to sell: 5
Sold 5 units of Backdrop
Enter item ID to sell (0 to stop): 0

Updated Inventory:
ID: 1, Name: Camera, Quantity: 10, Price: $887.00
ID: 3, Name: Backdrop, Quantity: 10, Price: $234.00
ID: 5, Name: Basket, Quantity: 9, Price: $100.00
gppavitharani@Its-Paviii LAB1-4 % 

```