**Arrays**

// Declare an array of 5 strings with each element initialized to its zero value.

//

// Declare a second array of 5 strings and initialize this array with literal string

// values. Assign the second array to the first and display the results of the first array.

// Display the string value and address of each element.

package main

import "fmt"

func main() {

// Declare string arrays to hold names.

var names [5]string

// Declare an array pre-populated with friend's names.

friends := [5]string{"Joe", "Ed", "Jim", "Erick", "Bill"}

// Assign the array of friends to the names array.

names = friends

// Display each string value and address index in names.

for i, name := range names {

fmt.Println(name, &names[i])

}

}

**Slices**

// Declare a nil slice of integers. Create a loop that appends 10 values to the

// slice. Iterate over the slice and display each value.

//

// Declare a slice of five strings and initialize the slice with string literal

// values. Display all the elements. Take a slice of index one and two

// and display the index position and value of each element in the new slice.

package main

import "fmt"

func main() {

// Declare a nil slice of integers.

var numbers []int

// Append numbers to the slice.

for i := 0; i < 10; i++ {

numbers = append(numbers, i\*10)

}

// Display each value.

for \_, number := range numbers {

fmt.Println(number)

}

// Declare a slice of strings.

names := []string{"Bill", "Joan", "Jim", "Cathy", "Beth"}

// Display each index position and name.

for i, name := range names {

fmt.Printf("Index: %d Name: %s\n", i, name)

}

// Take a slice of index 1 and 2.

slice := names[1:3]

// Display the value of the new slice.

for i, name := range slice {

fmt.Printf("Index: %d Name: %s\n", i, name)

}

}

**Maps**

// Declare and make a map of integer values with a string as the key. Populate the

// map with five values and iterate over the map to display the key/value pairs.

package main

import "fmt"

func main() {

// Declare and make a map of integer type values.

departments := make(map[string]int)

// Initialize some data into the map.

departments["IT"] = 20

departments["Marketing"] = 15

departments["Executives"] = 5

departments["Sales"] = 50

departments["Security"] = 8

// Display each key/value pair.

for key, value := range departments {

fmt.Printf("Dept: %s People: %d\n", key, value)

}

}