



Program Structure



```
package main
```

```
import <packages>
```

```
func main() {  
    statements  
} // main()
```



Pre-defined Types



- **Integers:** `uint8`, `uint16`, `uint32`, `uint64`, `int8`, `int16`, `int32`, `int64`, `int`
(size is machine-dependent 32 or 64 bit)
Operators: `+`, `-`, `*`, `/`, and `%`
- **Floating point numbers:** `float32` and `float64`
Operators: `+`, `-`, `*`, and `/`
- **Booleans:** `bool` (`false`, `true`)
Operators: `&&`, `||`, and `!`
- **Strings:** `string`. E.g. `"Bryn Mawr"`, `"Haverford"`, etc.
Operations: `+` (concatenation), `s[i]` (`i`-the char `s`), and `len(s)`
Strings can be compared using `<`, `<=`, `>`, `>=`, `==`, and `!=`.



Variables



- **Basic:** `var <name> <type>`
- **Initializing variables:** `var <name> <type> = <value>`
- **Type Inference:** `var <name> = <value>`
- **Multiple variables:** `var <name1>, <name2>, ... <type>`
- **Group Definitions:**
- **Short-hand Declarations:** `<var> := <value>`
- All variables are initialized by default: 0 for integers, 0.0 for floats, "" for strings, and false for Boolean.

```
var n int
n int = 10
var n = 10
var n, m int
var n, m = 10, 20
(
    n = 10
    name = "Bryn Mawr "
    rank int
)
n := 10
```



Control Flow - Assignment



- **Assignment:** `<var> = <expression>` `area = width * height`
- **Multiple Assignment:**

```
<var1>, <Var2>, ..., <varN> = <expr1>, <expr2>, ..., <exprN>  
<var1>, <Var2>, ..., <varN> := <expr1>, <expr2>, ..., <exprN>
```

```
n, m := 10, 20
```

Multiple assignment can also be used to swap two variables:

```
n, m = m, n
```

Functions can return multiple values where multiple assignment can also be used.



Control Flow - Conditional



- **Conditional Statements-If**

```
if <condition> {  
    <statement(s)>  
}
```

```
max = b  
if a > b {  
    max = a  
}
```

```
if <condition> {  
    <statement(s)>  
} else {  
    <statement(s)>  
}
```

```
if a > b {  
    max = a  
} else {  
    max = b  
}
```



Conditional – Switch/Case



```
switch <expression> {  
  case <expression1> : <statement(s)>  
  case <expression2>: <statement(s)>  
  case <expression2>: <statement(s)>  
  ...  
  default: <statement(s)>  
}
```

```
switch month {  
  case 1, 3, 5, 7, 8, 10, 12:  
    daysInMonth = 31  
  case 4, 6, 9, 11:  
    daysInMonth = 30  
  case 2: if (leapYear(year) {  
    daysInMonth = 29  
  } else {  
    daysInMonth = 28  
  }  
  default: fmt.Printf(ERROR)  
    os.Exit(1)  
}
```



Control Flow - Loops



```
for <init>; <condition>; <post> {  
    <statements>  
}
```

```
for i := 0; i < n; i++ {  
    sum += i  
}
```

For as a while loop:

```
for a != b {  
    if a > b {  
        a = a - b  
    } else {  
        b = b - a  
    }  
}
```

Looping through an array, A

```
for i, x := range A {  
    fmt.Printf("index:%d, element: %v\n", i, x)  
}
```