



Laboratorio di  
program-  
mazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

# Laboratorio di programmazione

## Corsi di laurea triennale unificati

### Turno A (A-Cao)

Docente: Anna Morpurgo  
Tutor: Umberto Costantini

Dipartimento di Informatica  
Università degli Studi di Milano

A.A. 2019-2020



- **aula:** 309, Settore didattico
- **sito web:**  
<http://morpurgo.di.unimi.it/didattica/LabProgrammazione/>
- **inizio:** 10/10/2019
- **fine:** 9/01/2020
- **ricevimento:** per appuntamento (da chiedere a lezione o via mail)
- **appelli d'esame:** gennaio, febbraio, giugno, luglio, settembre, gennaio

<https://www.youtube.com/watch?v=2qkRrWlwA08&feature=youtu.be>



**Finalità:** individuare le principali difficoltà incontrate dagli studenti nel corso di programmazione, proporre alcuni interventi mirati e individuare quali hanno impatto, e in che misura, sul superamento dell'esame.

## Dati:

- la frequenza a lezione  
(<https://easystaff.divisi.unimi.it/PortaleStudenti/>)
- le risposte alle domande di quiz proposti a lezione
- le soluzioni di esercizi proposti in classe
- le soluzioni di esercizi proposti agli esami

# Laboratorio 02 - Programma svolto a teoria



Laboratorio di  
program-  
mazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

- 9/10/19: 3. Il primo programma in Go. Il go tool.  
Compilazione. Esecuzione. Formattazione.  
Documentazione
- 11/10/19: 4. Discussione degli aspetti lessicali e sintattici.  
Commenti. Struttura generale di un programma  
go: programma, pacchetti, sorgenti. La libreria  
standard.
- 14/10/19: 5. Variabili: nome, tipo, valore, scope. Tipi.  
Classificazione dei tipi (tipi di base, tipi composti,  
interfacce). Dichiarazione, assegnamenti e  
assegnamenti multipli, short-assignment.
- 16/10/19: 6. I/O di base: `fmt.Println`, `fmt.Print`, `fmt.Scan`. Tipi  
di base numerici (`int`, `float64`). Espressioni  
numeriche. Conversioni. Variabili inutilizzate e  
blank variable.

Quando si parla di un **linguaggio**, si distinguono **diversi livelli**:

- **lessicale**: simboli e termini
- **sintattico**: regole di formazione (di una frase corretta)
- **semantico**: significato
- **pragmatico**: uso

Esempio: “Sai che ore sono?”

- lessicale: {sai, che, ore, sono, ?}
- sintattico: {verbo, pronome, sostantivo, verbo}
- semantico: il significato letterale della frase
- pragmatico: lo scopo, cioè chiedere che ore sono

# Competenze per programmare



Laboratorio di  
programmazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

Tre livelli:

- **Conoscenza sintattica:** conoscenza della sintassi del linguaggio di programmazione.
- **Conoscenza concettuale:** conoscenza della dinamica del programma in esecuzione (*notional machine*).
- **Conoscenza strategica:** capacità di applicare la conoscenza sintattica e concettuale al fine di risolvere nuovi problemi e raggiungere obiettivi specifici.

Che corrispondono a:

- **codice sorgente**
- **programma in esecuzione**
- **scopo del programma**

# Il processo di sviluppo di un programma



Laboratorio di  
program-  
mazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

**Introduzione**

Strumenti

Introduzione al  
linguaggio

Esercizi

1. analisi delle specifiche

2. progettazione

→ 3. scrittura di codice e documentazione

4. compilazione

5. [correzione di errori sintattici]

6. *Testing*

7. [*debugging* (individuazione di errori logici e runtime)]

Nel progettare e scrivere faremo uso di *goals* e di *plans* e *patterns*.

Un esempio



<b>go build</b> hello_world.go	COMPILAZIONE
<b>./hello_world</b>	ESECUZIONE
<b>go run</b> hello_world.go	ESECUZIONE (crea eseguibile temporaneo)
<b>go fmt</b> hello_world	FORMATTAZIONE (modifica il file)
<b>gofmt</b> hello_world	FORMATTAZIONE (visualizza il risultato)





- struttura di un programma Go
- commenti
- variabili e costanti, assegnamento
- tipi int e float64
- Print e Println
- Scan
- operatori aritmetici

# Struttura di un programma Go



Laboratorio di  
program-  
mazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

```
//hello_world.go
/*
My first Go program
to greet the world
*/
package main

import "fmt"

func main() {
    fmt.Println("Hello World!")
}
```



Tutti i linguaggi di programmazione usano *identificatori* per dare nomi alle diverse entità utilizzate in un programma, tra cui:

- variabili
- costanti
- funzioni
- tipi definiti dal programmatore (*user defined types*)

## ***Name Use Pattern:***

- *declare*
- *define/build/initialize*
- *use*
- *destroy*



# L'istruzione var

Laboratorio di  
program-  
mazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

L'istruzione var **dichiara** una lista di variabili; il tipo è alla fine.

```
var i int
```

Un'istruzione var permette anche di **inizializzare** le variabili:

```
var i, j int = 1, 2
```

Se una variabile è inizializzata, il tipo può essere omesso; la variabile assumerà il tipo corretto, inferito dal valore nella parte destra

```
var n, correct, name = 3, false, "Paolo"
```



Variabili dichiarate senza un valore iniziale esplicito vengono inizializzate con il loro **valore zero**.

Il valore zero è:

- 0 per tipi numerici,
- false per il tipo booleano,
- "" (stringa vuota) per le stringhe



Le costanti sono dichiarate e inizializzate come le variabili, ma con la parola chiave `const`. Esempio:

```
const Pi = 3.14
```

Il loro valore **non può essere modificato**.

Le costanti possono essere carattere, stringa, booleano, o valori numerici.

Laboratorio di  
program-  
mazione

Introduzione al  
linguaggio

## Esercizi

- ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡



In Go operazioni o assegnamento tra elementi di tipo differente richiedono una **conversione esplicita**.

L'espressione  $T(v)$  converte il valore  $v$  nel tipo  $T$

```
var i = 42
var x = float64(i) //conversione per l'assegnamento
var y float64
y = x + float64(i) // conversione in un'espressione
```



# import e nomi pubblici



Laboratorio di  
program-  
mazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

Un programma deve **importare** i **package** di cui ha bisogno (es. `fmt`) tramite percorsi di import

```
import "fmt"
```

In Go, un **nome** è **pubblico** (esportabile) se comincia con una **lettera maiuscola**.

Quando si importa un package per usare le sue variabili o funzioni pubbliche, queste avranno sempre un nome che inizia con una maiuscola.

Ad esempio Print in:

```
fmt.Print
```

Per **scrivere** su standard output: funzioni **fmt.Print** e **fmt.Println** (fmt.Println aggiunge spazi e un a-capo alla fine).

Ad esempio

```
fmt.Print("Hello", 1)
fmt.Println("Hello", 2)
fmt.Print("Hello", 3)
```

produce in output:

**Hello1Hello 2**

**Hello3**

*Nota.* Per stampare più valori, separarli con virgole.

Per **leggere** da standard input: la funzione **fmt.Scan**. Ad esempio:

```
fmt.Println("Please enter your full name: ")
fmt.Scanln(&firstName, &lastName)
```



**Convenzione per input e output:** input e **output**

**Convenzione per i nomi dei file:** Chiamare i file dei programmi con il nome indicato preceduto dal numero della lezione. In particolare la convenzione è: **leznn\_nomefile.go**, dove *nn* è il numero della lezione di Go (la numero 01 la settimana dell'8 ottobre) e *nomefile* è il nome indicato, indicativo del problema proposto, ad esempio **lez01\_area\_cerchio.go**.

**Sito di upload** (per caricare i programmi):

<https://upload.di.unimi.it/>

corso: 2019 - LAB PROGRAMMAZIONE - GO - TURNO A  
(MORPURGO)

# Esercizio 1 - Operazioni aritmetiche



Laboratorio di  
program-  
mazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

## I/O e operatori aritmetici

*Problema:* Scrivere un programma Go operazioni.go che, dati due numeri float64, esegua addizione, sottrazione, moltiplicazione e divisione dei due numeri.

*Esempio di esecuzione:*

**inserisci due numeri (float) 4.5 2.1**

**somma = 6.6**

**differenza = 2.4**

**prodotto = 9.4500000000000001**

**quoziente = 2.142857142857143**

# Esercizio 2 - Quoziente e resto



Laboratorio di  
program-  
mazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

## I/O e operatore modulo

*Problema:* Scrivere un programma Go `quoziente_resto.go` che, dati un dividendo e un divisore (interi), calcoli il quoziente e il resto.

*Annotazioni* L'operatore per la divisione (`/`) tra interi calcola la parte intera del risultato. L'operatore per il resto della divisione è `%`.

*Esempio di esecuzione:*

**dividendo?**

2500

**divisore?**

235

**quoziente = 10**

**resto = 150**

# Esercizio 3 - Angoli di un triangolo



Laboratorio di  
program-  
mazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

## Uso di costanti, approssimazioni

*Problema:* Scrivere un programma Go `angolo_triangolo.go` che, date le ampiezze di due angoli di un triangolo, determini l'ampiezza del terzo angolo.

*Esempio di esecuzione:*

```
inserisci le ampiezze di due angoli di un triangolo
25.3 90.2
ampiezza terzo angolo: 64.49999999999999
```

# Esercizio 4 - Conversione da gradi centigradi a Fahrenheit



Laboratorio di  
program-  
mazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

## Conversioni tra grandezze, divisione tra int e tra float

*Problema:* Scrivere un programma Go `centigradi_fahrenheit.go` che converta in Fahrenheit una temperatura espressa in gradi centigradi.

Come si comporta il programma se dichiariamo le variabili come `int`? e come `float64`?

*Annotazioni.* Un grado Fahrenheit è  $\frac{5}{9}$  di un grado centigrado e 0 gradi centigradi corrisponde a 32 gradi Fahrenheit. Usare costanti nella formula di conversione.

*Esempio di esecuzione:*

**temperatura in centigradi?**

45

**45 C = 113 F**

# Esercizio 5 - Conversione da giorni in anni, settimane, giorni



Laboratorio di  
program-  
mazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

## Conversioni tra grandezze

*Problema:* Scrivere un programma Go `conversione_giorni.go` che converta un numero specificato di giorni (fornito dall'utente) in anni, settimane, giorni. Si ignorino gli anni bisestili.

*Indicazione.* Usare costanti per il numero di giorni (365) in un anno e il numero di giorni in una settimana.

*Esempio di esecuzione:*

**numero di giorni da convertire?**

1329

**1329 giorni equivalgono a 3 anni, 33 settimane, 3  
giorni**





# Esercizio 6 - Distanza tra due punti

## Uso del package "math"

*Problema:* Scrivere un programma Go `distanza_punti.go` che calcola la distanza tra due punti nel piano cartesiano.

*Annotazioni* Siano  $P_1 = (x_1, y_1)$  e  $P_2 = (x_2, y_2)$ .

La formula della distanza tra due punti  $P_1$  e  $P_2$  è:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

Siano  $x_1, y_1, x_2, y_2$  tutti valori float64.

*Suggerimenti.* La funzione per calcolare la radice quadrata si chiama `Sqrt` ed è messa a disposizione nel package "math".

*Esempio di esecuzione:*

**x e y del primo punto: 5 8**

**x e y del secondo punto: 7 11**

**La distanza tra i due punti è: 3.605551275463989**

Laboratorio di  
programmazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

# Esercizio 7 - Consumo medio e resa di un motore



Laboratorio di  
program-  
mazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

## Calcolo di grandezze

*Problema.* Scrivere un programma Go `consumo_resa.go` che calcola il consumo medio e la resa di un motore data la distanza totale percorsa (in km) e la quantità di carburante utilizzata (in litri). I valori sono di tipo `float64`.

*Annotazioni..* Il consumo medio di carburante si esprime in l/km ed è la quantità di carburante che occorre in media per percorrere un km di strada. La resa di un motore è data dalla distanza percorsa in media con un litro di carburante e si esprime in km/l.

*Esempio di esecuzione*

**distanza percorsa (in km): 50**

**quantità di carburante utilizzata (in l): 10**

**consumo medio: 0.2 l/km**

**resa media: 5 km/l**

# Esercizio 8 - Conversione da secondi in giorni, ore, minuti, secondi



Laboratorio di  
program-  
mazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

## Conversioni di grandezze

*Problema.* Scrivere un programma Go `conversione_secondi.go` che converta un numero dato di secondi (fornito dall'utente) in giorni, ore, minuti, secondi.

*Esempio di esecuzione*

**numero di secondi da convertire:**

123456

**g:h:m:s - 1:10:17:36**

# Esercizio 9 - Intersezione di due rette



Laboratorio di  
program-  
mazione

A. Morpurgo

Info generali

Laboratorio 02

Programma

Introduzione

Strumenti

Introduzione al  
linguaggio

Esercizi

## Problema e sottoproblemi

*Problema.* Scrivere un programma Go `intersezione rette.go` che, date le equazioni di due rette, stabilisce in che punto si intersecano.

*Annotazioni..* L'equazione della retta è:  $y = mx + q$

*Esempio di esecuzione*

**retta 1: m e q? 1 4**

**retta 2: m e q? 2 6**

**intersezione in (-2,2)**