

PROGRAMMAZIONE 1 E LABORATORIO

PAOLO MILAZZO (milazzo@di.unipi.it)

<http://pages.di.unipi.it/milazzo/PRL>

MAIUSCOLO

PARTE TEORICA (9 CFU
~ 72 ore)

LABORATORIO (3 CFU
~ 24 ore)

VEDERE LA PAGINA WEB
DEL LAB. PER LA SUDDIVISIONE
IN GRUPPI

RICEVIMENTO DOCENTE

(Dipartimento di Informatica)

ORARIO: MARTEDÌ 14.00-16.00

(MEGLIO SE MI CONTATTATE
PRIMA VIA EMAIL)

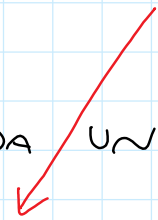
ESAME :

- PUO' ESSERE SOSTITUITA DAL SUPERAMENTO DELLE VERIFICHE INTERMEDIE (N INIZIO NOVEMBRE E PRIMA DI NATALE)
- PROVA SCRITTA (ESERCIZIO)
 - PROVA PRATICA IN LABORATORIO
 - PROVA ORALE (DOMANDE DI TEORIA)

PROGRAMMA :

DESCRIZIONE ESEQUIBILE DA UN
CALCOLATORE DI UN METODO PER
IL CALCOLO DI UN RISULTATO A
PARTIRE DA DATI INIZIALI

ALGORITMO



INPUT



OUTPUT

PARADIGMI DI PROGRAMMAZIONE

PROGRAMMAZIONE IMPERATIVA
(PROGRAMMA = SEQUENZA
di comandi)

LINGUAGGI
DI PROGRAMMAZIONE

C

NEI
NOSTRO
CORSO
✓

PROGRAMMAZIONE FUNZIONALE
(PROGRAMMA = FUNZIONI)

CAML

✓

PROGRAMMAZIONE A OGGETTI
(PROGRAMMA = INSIEME di
ELEMENTI OGGETTI di
VARIE FUNZIONALITÀ)

JAVA

C++ C#

PROGRAMMAZIONE
2

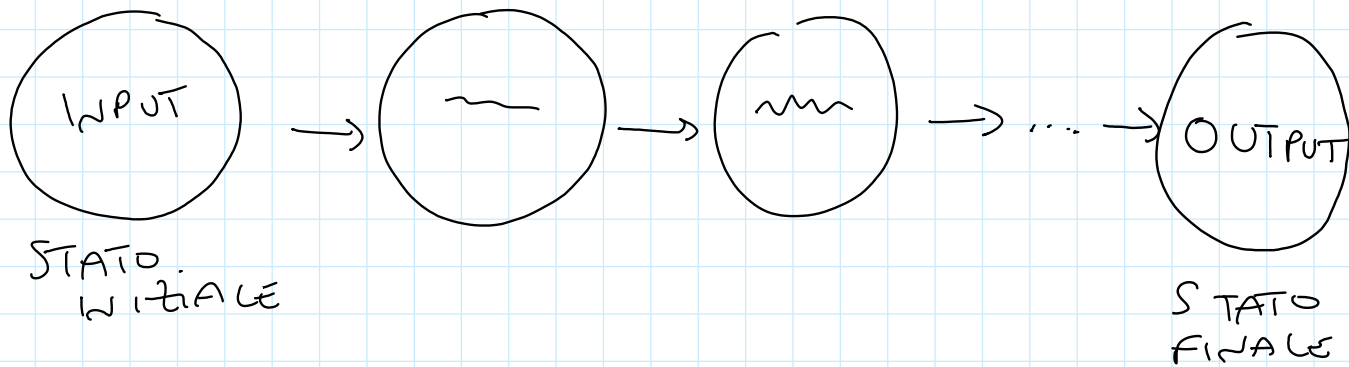
⋮
⋮
⋮

ARGOMENTI DEL CORSO

- PROGRAMMAZIONE IMPERATIVA IN C
- SINTASSI DEI LINGUAGGI DI PROGRAMMAZIONE
(GRAMMATICHE)
- TEORIA DELLA RICORSIONE
(INTRODUZIONE ALLA PROGRAMMAZIONE FUNZIONALE)
- PROGRAMMAZIONE FUNZIONALE
IN CAML

PROGRAMMAZIONE IMPERATIVA

martedì 19 settembre 2017 11:33



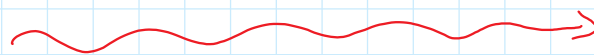
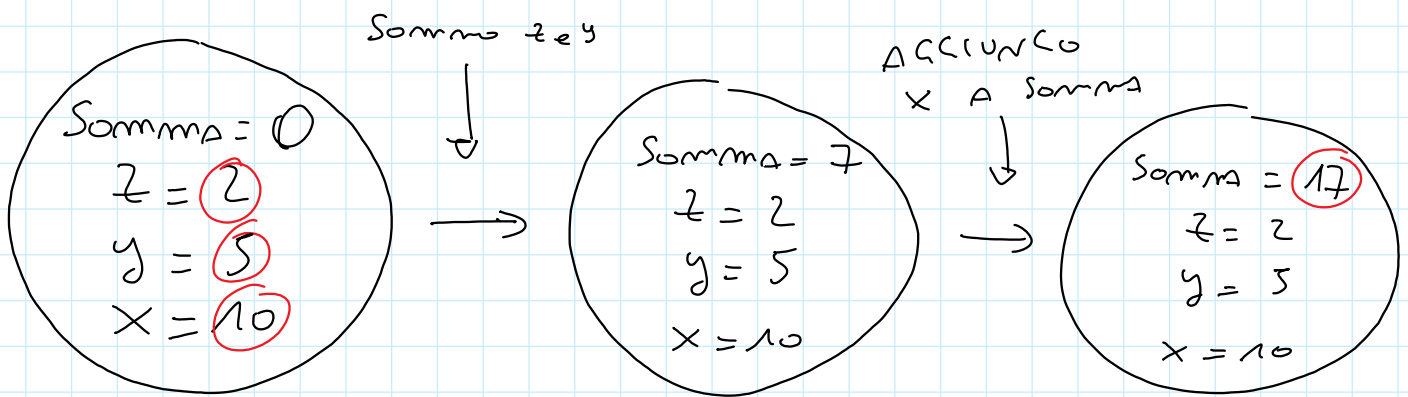
UNO STATO È DATO DA UN
INSIEME DI VARIABILI

VARIABILE = UN NOME CHE È ASSOCIATO AD UN VALORE
(N ALLE INCOGNITE IN ALGEBRA)

IL VALORE DI UNA VARIABILE PUÒ ESSERE MODIFICATO

ESEMPIO: SOMMA DI TRE NUMERI

x=10 y=5 z=2 Somma



ESECUZIONE DEL
PROGRAMMA

$$3 - x + y$$

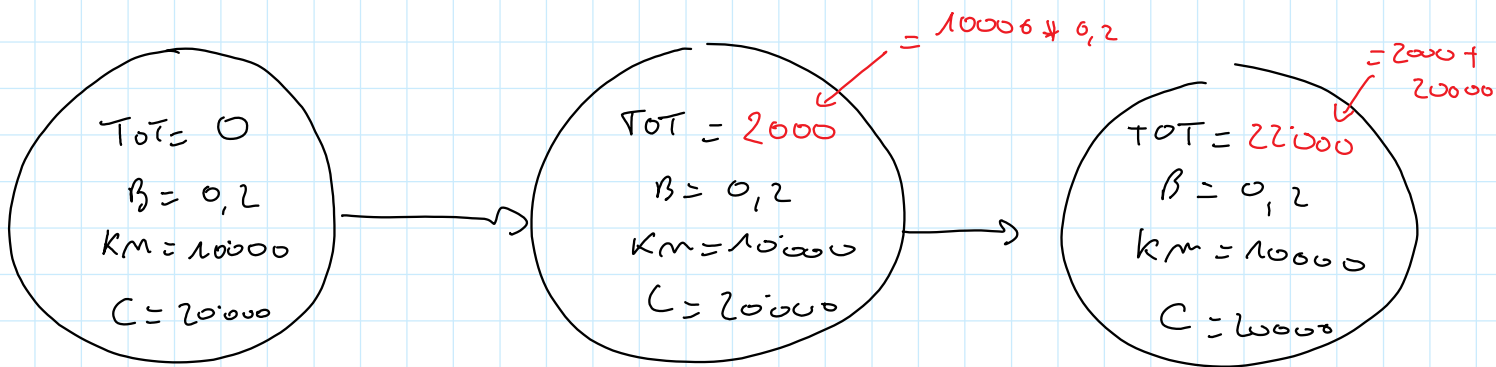
ESEMPIO :

martedì 19 settembre 2017 11:52

CON 30'000 € COPRIRE LE
SPESE DI ACQUISTO DI UN'AUTO
E IL CARBURANTE PER 1 ANNO

DATI : COSTO AUTO = 20'000 € \checkmark C
KM PERCORSI = 10'000 km \checkmark km
COSTO BENZINA = 0,20 € \checkmark B
AL KM

OUTPUT : COSTO TOTALE \checkmark TOT



PROGRAMMA

$$TOT = KM * B;$$

$$TOT = TOT + C;$$

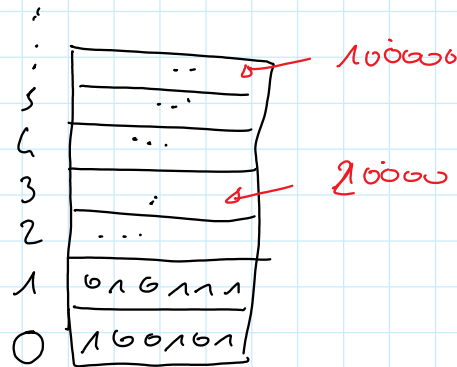
\uparrow
SCRIVENDO
UN NUOVO
VALORE IN
TOT

\uparrow
LEGO IL
VECCHIO VALORE
DI TOT (2000)

}o TOT È RICALCOLATO
IN TERMINI DI
SE STESSO

RAPPRESENTAZIONE DELLO STATO

DI UN PROGRAMMA NELLA MEMORIA
DI UN COMPUTER

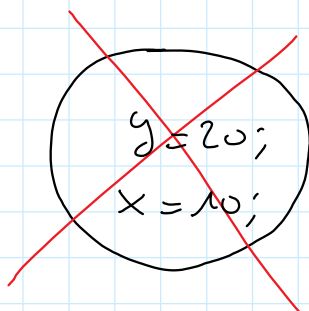


INDIRIZZI DI MEMORIA
(O LOCALITÀ)

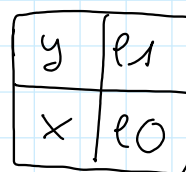
SERVE UNO
STRUMENTO
CHE MI DICA
A QUALE
di
MEMORIA
CORRISPONDE
OGNI VARIABILE

AMBIENTE

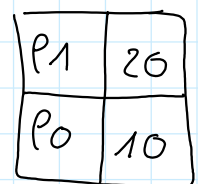
LO STATO È IN REALTÀ
COSTITUITO DA UN
AMBIENTE + UNA MEMORIA



=>



AMBIENTE



MEMORIA

lo rappresenta P_c
posizione di memoria
con indice θ in cui
è memorizzato il
valore di x

DICHIARAZIONI DI VARIABILI

martedì 19 settembre 2017 12:26

→ SERVONO PER COSTRUIRE UNO STATO

SINTASSI:

- TIPO VARIABILE ;

DICHIARAZIONE
SEMPLICE

- TIPO VARIABILE = ESPRESSIONE ;

DICHIARAZIONE
CON
INIZIALIZZAZIONE

TIPO =
INFORMAZIONE
SU QUALI VALORI
POTRANNO ESSERE
ASSOCIATI ALLA
VARIABILE

INT (NUMERI
INTERI)

FLOAT (NUMERI
FRAZIONARI)

CHAR (CARATTERI)

int km ;



DICHIARAZIONE
SEMPLICE, NON
ASSOCIA NESSUN
VALORE INIZIALE
ALLA VARIABILE

int km = 10000 ;



LA VARIABILE
VIENE CREATA
CON VALORE
INIZIALE
10000

→ LE VARIABILI DEVONO
ESSERE DICHIARATE PRIMA
DI POTER ESSERE USATE

→ OGNI VARIABILE DEVE ESSERE
DICHIARATA UNA SOLA VOLTA

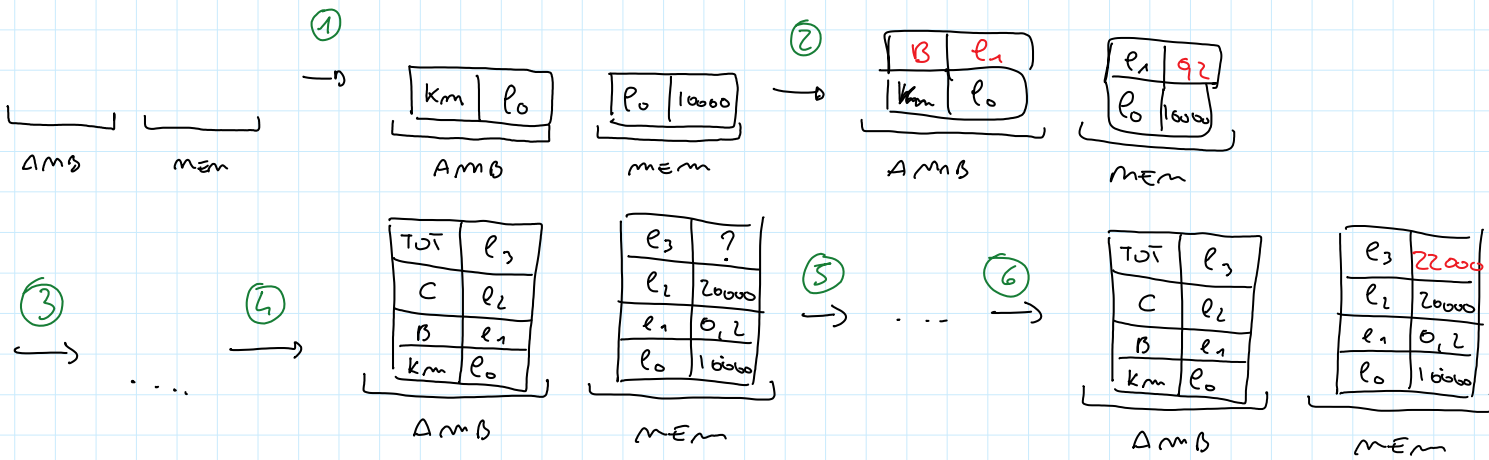
AGGIUNGIAMO LE DICHIARAZIONI

martedì 19 settembre 2017 12:37

```

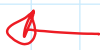
int Km = 10000;
float B = 0,2;
int C = 20000;
float Tot;
Tot = Km * B;
Tot = Tot + C;
    
```

- ①
- ②
- ③
- ④
- ⑤
- ⑥



ATTENZIONE

```
int x;  
int y = 10;
```

y = y + x;  LA VARIABILE X NON ERA STATA INIZIALIZZATA

BISOGNA SEMPRE ASSICURARSI CHE ALLE VARIABILI USATE IN UN'ESPRESSIONE SIA STATO ASSOCIATO IN PRECEDENZA UN VALORE

DI CHIAMATA
CON INIZIALIZZAZIONE

```
int x = 10
```

```
⋮  
y = x;
```

✓

PRECEDENTEMENTE
ASSEGNOTA

```
int x;
```

```
⋮  
x = 10  
⋮  
y = x;
```

✓