WELCOME TO C++

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

MODALITA' D'ESAME (iscriversi entro una settimana prima)

Esame solo scritto (massimo 33 punti):

- prima prova: (10 punti circa): domande a crocette (risposta sbagliata o non data vale 0) sugli argomenti spiegati dalla Prof. Vaglini e forse anche su C++.
- seconda prova (23 punti circa): 3 esercizi in C++ sui vettori di strutture, matrici e liste.

Limite di 4 consegne in un anno

Per definire sintassi e semantica di un linguaggio occorre utilizzare un altro linguaggio, ossia un *metalinguaggio*

•Metalinguaggio per la sintassi C++: insieme di notazioni (non ambigue), che possono essere spiegate con poche parole del linguaggio naturale.

•Metalinguaggio per la semantica C++: assai complesso, si ricorre direttamente al linguaggio naturale.

•Notazione utilizzata per la sintassi C++: derivata dal classico formalismo di Backus e Naur (*BNF*, *Backus-Naur Form*).

La notazione utilizzata è basata sulla grammatica BNF, con terminologia inglese, regole semplificate rispetto alla sintassi ufficiale, caratterizzato dal prefisso basic e diversa organizzazione delle categorie sintattiche.

Regola: descrive una categoria sintattica utilizzando altre categorie sintattiche, costrutti del metalinguaggio e simboli terminali.

Una sintassi corretta non implica una semantica corretta.

Gli elementi di una categoria sintattica possono essere opzionali (contrassegnati con il suffisso |opt) e possono essere ripetuti più volte (mediante categorie sintattiche aggiuntive).

Un **programma C++** è costituito da sequenze di parole (token) che possono essere delimitate da spazi bianchi (whitespace).

Le parole sono costituite dai seguenti caratteri: token character, digit (0 1 2 …9) letter (a b.. A B…), special (! & % //…)

→Spazi bianchi: carattere spazio, carattere tabulazione (orizzontale e verticale), carattere *nuova riga e ritorno carrello*.

→Commenti:

–sequenze di parole e spazi bianchi racchiuse fra i caratteri /* e */, oppure fra i caratteri // e la fine della riga;

–hanno lo scopo di documentare un programma;

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

–possono essere inseriti liberamente nel testo e non hanno alcun effetto sull'esecuzione del programma.
Gli spazi bianchi e i commenti costituiscono le spaziature.

- **Categorie sintattiche elementari (elementi lessicali):**

–opportune sequenze di caratteri (token-character o whitespace);
–non possono includere spaziature (aggiuntive) fra un carattere e un altro.

- **Elementi lessicali:**

–identificatori (identifier);
–parole chiave (keyword);
–espressioni letterali (literal);
–operatori (operator);
–separatori (separator).

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

Le entità usate in un programma devono possedere nomi e i nomi possono essere identificatori (identifier, letter, identifier char, digit). I termini nome e identificatore spesso vengono usati intercambiabilmente, ma è necessario distinguerli:
»un nome può essere un identificatore, oppure un identificatore con altri simboli aggiuntivi.
Il carattere di sottolineatura _ è una lettera e la doppia sottolineatura all'interno degli identificatori è sconsigliata perché riservata alle implementazioni e alle librerie.
Il c++ distingue tra minuscole e maiuscole (è case sensitive).

- **Parole chiave (keyword):**

–simboli costituiti da parole inglesi (formate da sequenze di lettere), il cui significato è stabilito dal linguaggio: and ... while ...for
Un identificatore non può essere uguale ad una parola chiave.

- **Espressioni letterali:**

–chiamate semplicemente letterali;
–denotano valori costanti (costanti senza nome);
» numeri interi (per es. 10);
» numeri reali (per es. –12.5);
» letterali carattere (per es. 'a');
» letterali stringa (per es. "informatica").

- **Operatori:** sono caratteri speciali e loro combinazioni, servono a denotare operazioni nel calcolo delle espressioni. (+,- , * ecc) Esiste quindi in libreria un operatore + e si può sfruttare la proprietà associativa di questa funzione come avviene in algebra.

  Proprietà degli operatori:

- posizione rispetto ai suoi operandi (argomenti) : prefisso (precede gli argomenti, es. +5), postfisso (segue gli argomenti, es. x++), infisso (in tutti gli altri casi, es. 4+5).
- numero di argomenti (arietà)
- precedenza (o priorità) consultata dal compilatore qualora nell'espressione vi fossero diversi operatori : nell'ordine di esecuzione: gli operatori con priorità più alta vengono eseguiti per primi; es. : arg1 + arg2 * arg3 (operatore prodotto ha priorità maggiore dell'operatore somma) . Se scrivo "cout<<3+5*4;" compaiono due operatori distinti e a ogni operatore è associata una priorità: la priorità della moltiplicazione è maggiore della priorità dell'addizioni".
- associatività: servono almeno due operandi. Si definiscono:

operatori associativi a sinistra: vengono eseguiti da sinistra a destra: es. arg1+arg2+arg3 → (arg1+arg2)+arg3

operatori associativi a destra : vengono eseguiti da destra a sinistra: es. arg1=arg2=arg3 → arg1=(arg2=arg3)

- **Separatori:**

–simboli di interpunzione, che indicano il termine di una istruzione, separano elementi di liste, raggruppano istruzioni o espressioni:

Es. carattere punto e virgola (;), coppia di caratteri ( )

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

Il linguaggio C++ trasforma una sequenza di istruzioni scritte in un file di testo in istruzioni per la macchina dunque per il processore. Deve esistere un traduttore che possa tradurre il file di testo da codice ASCII in istruzioni valide per C++.

Questo compito è svolto dal **compilatore** C++ che prende in ingresso un file di testo o file sorgente e produce in uscita un file oggetto:

→ IN) p1.cpp // "file di testo in ASCII"

→ OUT) p1.0 // "file binario con codice oggetto che non è ancora eseguibile.

Il compilatore analizza completamente il codice prima di metterlo in esecuzione quindi è facile ottenere errori perché deve controllare la traduzione prima di metterla in esecuzione. Dopo la compilazione serve un'ulteriore fase che è la **fase di collegamento** (**linking**) che prende in ingresso un file binario in oggetto e produce in uscita un file binario eseguibile (in Windows hanno l'istruzione exe).

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

→ IN) p1.0 // "file oggetto"

→ OUT) "file binario eseguibile".

Bisogna analizzare i file sorgenti, applicare i linking e dopo lanciare il programma in esecuzione.

Iostream è la libreria che legge caratteri alfanumerici e permette di visualizzarne in video altri.

File eseguibile : sequenza di bit 0 e 1 salvati su disco. Prelevando un byte alla volta rappresenta un codice di istruzioni per un processore.

Programma C++ : sequenza di istruzioni ben formate e valide che dopo essere state tradotte e decodificare verranno eseguite in sequenza.

return 0 = è uno stop e indica il termine del programma e il controllo passa al sistema operativo.

La nomenclatura consentita in C++ non consente in modo semplice e conveniente di trattare separatamene dichiarazioni e definizioni, con dichiarazione si può intendere sia una dichiarazione vera e propria sia una definizione.

Il nucleo base di un programma è il seguente:

→ #include <cstdlib> //direttive per il preprocessore

```
#include <iostream>
using namespace
std;
int main ( )
{
return 0;
```

→ using namespace std; //direttiva che indica al compilatore che tutti i nomi usati nel programma si riferiscono allo standard ANSI-C++

→ int main() // dichiarazione della funzione main

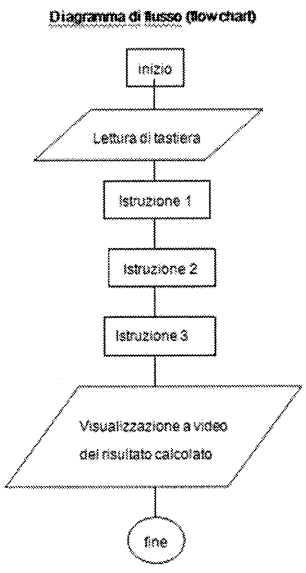
→ system("PAUSE"); // consente di bloccare l'esecuzione in attesa che l'utente digiti un tasto. La funzione PAUSE può essere invocata solo nel sistema operativo Windows

```
#include <iostream>
using namespace std;
int main ( )
{
cout<<"ciao
mondo"<<endl;
return 0;
}
```

Ad esempio il seguente programma mostra la scritta "ciao mondo" in una finestra di dialogo.

Le sequenze di byte vengono decodificate in caratteri mediante le tabelle ASCII. Notepad riesce a fare questa corrispondenza (Es. 10010001 potrebbe corrispondere a #).

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745



C'è un intera classe di algoritmi per la quale è sufficiente fornire la sequenza giusta di istruzioni C++ per arrivare ai risultati: questa famiglia è costituita da algoritmi sequenziali da eseguire in ordine. Un algoritmo di questo tipo ha un corrispettivo in C++ del genere:

```
int main ()
{
Istr_1
Istr_2

Istr_n
return 0; }
```

N.B. Le istruzioni devono terminare col punto e virgola ;

int sta per intero, non razionale.

cout <<"...."<< stampa il risultato a video

cin >>"....">> legge numericamente l'ingresso dalla tastiera

La cin deve aspettare che l'utente inserisca i caratteri e appena si preme invio i caratteri sono convertiti in complemento a 2, salvati nel contenitore ad esempio "base" e la libreria fa la conversione da ASCII in binario su 32 bit.

L'indicatore *cout* e *cin* sono definiti a partire dalla libreria di ingresso e di uscita, cioè iostream.

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

Un programma comunica con l'esterno tramite uno o più flussi (stream). Lo **stream** è una struttura logica costituita da una sequenza di caselle ciascuna di un byte che termina con una marca di fine stream (il numero delle caselle è illimitato).

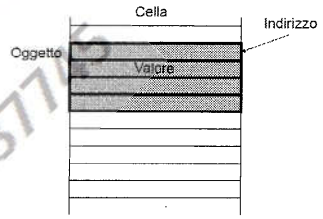
Gli stream predefiniti sono tre: cin, cout, cerr. Le funzioni che operano su questi stream si trovano in una libreria di ingresso/uscita e per il loro uso occorre includere il file di intestazione <iostream>.

Cin è lo stream di ingresso standard per prelevare i dati. L'operatore di lettura viene definito per singoli caratteri, numeri interi, reali e sequenze di caratteri (stringa). Quando viene incontrata un'istruzione di lettura il programma si arresta in attesa dei dati che verranno battuti a tastiera a riempire lo stream *cin* . Tali caratteri vanno effettivamente a far parte di cin solo quando viene battuto il tasto di ritorno carrello.

Cout è lo stream di uscita standard che permette di scrivere numeri interi, reali, caratteri, sequenze di caratteri (stringaa). Un valore booleano viene implicitamente convertito in intero.

Memoria: insieme di celle. **Cella**: in genere dimensione di un byte (8 bit)

Oggetto: gruppo di celle consecutive che vengono considerate dal programmatore come un'unica cella informativa.



Attributi di un oggetto: Indirizzo della prima cella e Valore (contenuto di tutte le celle).

Oggetti costanti (costanti con nome) e oggetti variabili:

–l'indirizzo comunque non cambia;

–il valore non può o può subire modifiche, rispettivamente.

Come viene vista la RAM (memoria centrale) dal linguaggio C++?

Nella memoria del disco rigido (hard disk) dove si salvano documenti, file ecc. = l'hard disk è una memoria persistente, e i dati salvati rimangono anche senza alimentazione elettrica. Nella memoria centrale (RAM) un programma è eseguibile più velocemente e col passare degli anni la sua capacità è aumentata.

Quindi in un computer abbiamo:

-CPU: legge e scrive dati nella memoria centrale e anche nella RAM. Interagisce con la tastiera e con il video, che sono periferiche. Il video è una periferica di sola uscita, la tastiera è una periferica di sola entrata. Il disco è una periferica di entrata e uscita.

-RAM

-Hard Disk

Nella maggior parte del tempo di esecuzione vi è l'interazione tra CPU e Ram.

La **RAM** viene vista come una sequenza di celle di 8 bit (=1 byte) e in base alla capienza della RAM, il numero di celle aumenta.