



Tecniche di Programmazione

(AA-JY)

01FYZOA

A.A. 2021/2022

INTRODUZIONE AL CORSO



Teachers

Lezioni:

- Prof. **Gianpiero Cabodi**, e-mail: gianpiero.cabodi@polito.it
- Prof. **Paolo Camurati**, e-mail: paolo.camurati@polito.it

Laboratori:

- Ing. **Marco Castangia**, e-mail: marco.castangia@polito.it
- Ing. **Federico Manuri**, e-mail: federico.manuri@polito.it

Contenuti del corso

- Secondo corso, per gli studenti di Ing. Informatica, nell'introduzione alla programmazione come strumento di problem solving, applicato a problemi (informatici) reali
- Consolidamento di attitudine al problem solving, conoscenza e capacità di realizzare strutture dati e algoritmi
- Introduzione a un nuovo linguaggio di programmazione: C
- Approccio in parte diverso dal corso di Python: **learning by experience and imitation**, meno tempo dedicato a dettagli sintattici (che in parte possono essere appresi individualmente e facendo riferimento a materiale esistente...)

Descrizione dettagliata sul portale (sezione «guida»):

https://didattica.polito.it/pls/portal30/gap.pkg_guide.viewGap?p_cod_ins=01FYZOA&p_a_acc=2022&p_header=S&p_lang=&multi=N

Contenuti del corso

- Nozioni base di architettura HW e logica
- C come secondo linguaggio di programmazione
- Puntatori in C
- Problem-solving elementare

Materiale

- Slide del corso e materiale aggiuntivo reso disponibile tramite il Portale della Didattica
- Libro usato per il corso (contiene descrizione sintetica del C): G. Cabodi, P. Camurati, P. Pasini, D. Patti, D. Vendraminetto, "Dal problema al programma: introduzione al problem-solving in linguaggio C", Apogeo, II edizione, 2016
- Suggerito per il linguaggio C come testo di riferimento: Deitel & Deitel, "C: how to program", Pearson Prentice Hall, 2010

Materiale aggiuntivo:

- Slide, libri, video, tutorial reperibili su web, possono essere utili per approfondire gli argomenti del corso

Organizzazione pratica

| | Lunedì | Martedì | Mercoledì | Giovedì | Venerdì |
|-------------|-------------|---------------------------|-----------|---------|---------|
| 08:30-10:00 | | | | LEZIONE | |
| 10:00-11:30 | | | | LEZIONE | |
| 13:00-14:30 | LAB (SQ. 1) | | | | |
| 14:30-16:00 | | LAB (SQ. 2) | | | |
| 16:00-17:30 | | LAB (SQ. 3) LEZIONE(*) | | | |

LAB - Sq 1: AB-CAP / **Sq 2:** CAR-DE (+ student/esse senza Notebook personale) / **Sq 3:** DI-JO

4.5 ore/sett, 14 settimane totali. 10 laboratori. Corso in presenza, lezioni pre-registrate disponibili. Laboratori dalla 2ª settimana di Corso. Prima settimana, installazione guidata di Clion

(*) **Solo in caso di recupero o no lab**, lezione extra il Martedì

Prima settimana no lab → lezione Martedì 1/3/2022

Verificare sempre Portale della Didattica per possibili aggiornamenti/modifiche!

Frequenza remota

Lezioni pre-registrate

- Video e PDF disponibili sul portale della didattica
- Per domande:
 - Canali (tematici) su Gruppo di lavoro slack (TP1-Polito-2022)

Laboratori

- pubblicati su base settimanale (portale della didattica)
- si possono svolgere a casa e/o in laboratorio
 - Non è garantito che si riesca a fare tutto in 1h30 (includono lavoro di studio/esercizio personale)
 - La presenza in laboratorio è utile per interagire con gli assistenti
 - Possono essere (in parte) consegnati ed essere valutati (spiegato successivamente)

Per comunicazioni off-line

- PORTALE DELLA DIDATTICA e e-mail Smatricola@studenti.polito.it
- Come strumento aggiuntivo (per comunicazioni in stile chat, e come supporto per le attività di laboratorio, si è creato un **gruppo di lavoro Slack per il corso**: tp1-polito-2022.slack.com
 - Attendere il link per l'iscrizione
 - Il link di Invito viene pubblicato sul portale
 - No acronimi: usare nome e cognome
 - No avatar, icona o simile: usare fotografia (tipo foto tessera)

Strumenti di programmazione

- Si può usare l'IDE (Integrated Development Environment) per programmazione C che si preferisce
- Si suggerisce CLion:
 - Un prodotto JetBrains (quindi simile a PyCharm)
 - Cross-platform (c'è per Windows, MAC e Linux)
 - Licenza Educational disponibile per students (la stessa di PyCharm)
 - Si assume che, dopo il corso Python, lo studente sia in grado di installare e configurare un IDE in modo autonomo, seguendo le istruzioni disponibili online:
<https://www.jetbrains.com/clion/>
 - Si fornirà tuttavia un tutorial mediante il Portale della Didattica
 - NON disponibile sui PC/LAIB (usare Codeblocks)



Esame

- Esame scritto (su carta), 90 min. che comprende:
 - Domande a risposta aperta e/o quiz a scelta multipla
 - Esercizi di Programmazione/problem solving in C
 - Maggiori dettagli/logistica/esempi saranno forniti più avanti, durante il corso
- Alla fine dell'esame si può fotografare l'elaborato (domande di programmazione). Si dovrà poi caricare, entro tre giorni, sul Portale (sezione elaborati):
 - Una breve relazione (max 1 pagina) che descriva, per gli esercizi di programmazione, la soluzione adottata (strutture dati, algoritmi, ...)
 - Una copia del programma corretto (deve poter essere compilato/eseguito in modo corretto), evidenziando modifiche e/o correzioni rispetto alla versione originale.
 - La non consegna del materiale richiesto viene intesa come ritiro dall'esame (non si ottiene valutazione)

Laboratori

- Gli **esercizi di laboratorio** possono essere consegnati e valutati
 - La valutazione può portare fino a 2 punti aggiuntivi sul voto finale di esame
 - Validità del laboratorio ai fini del punteggio: 28 febbraio 2023
- Le scadenze per le consegne saranno comunicate (di massima ci saranno 3 scadenze per 3 gruppi di laboratori)
- Si tratta di un extra: non è obbligatorio
 - Si consiglia COMUNQUE DI SVOLGERE I LABORATORI (anche se in ritardo) perché servono a preparare l'esame.

Consigli per affrontare il corso

- Non solo SAPERE (lezioni, comprensione, studio)
- Ma anche SAPER FARE (capire, applicare, esercizi, laboratori)
- **DAL SAPERE AL SAPER FARE:** per risolvere i problemi di programmazione:
 - atteggiamento **errato**: illudersi di risolvere un problema come se si ponesse per la prima volta, senza conoscere quanto già scoperto
 - atteggiamento **corretto**: conoscere e capire la teoria sottostante per poi applicarla.
- **SAPER FARE BENE:** non basta che il programma «funzioni»:
 - deve essere efficiente, leggibile, manutenibile, affidabile etc etc
- Si ricevono frequentemente domande del tipo

«devo .. posso... è lecito... è obbligatorio... è giusto/sbagliato...?»

La risposta è quasi sempre: **«puoi fare quello che ritieni più opportuno»**.

Uno degli obiettivi del corso è allenare alle «scelte responsabili», che rispettino i vincoli del problema, a volte accettando compromessi, ponendo enfasi su aspetti ritenuti importanti e magari rinunciando ad aspetti secondari. Le scelte fatte possono essere oggetto di valutazione, quasi sempre NON di tipo binario (SI/NO, ON/OFF, GIUSTO/SBAGLIATO)

I corsi di programmazione

Passato (fino al 2019/2020):

- Linguaggio C: Informatica (I anno - 8 crediti)
- Problem solving elementare e avanzato, ricorsione: Algoritmi e Programmazione (II Anno – 12 crediti)

Presente/future (dal 2020/2021)

- Programmazione in Python: Informatica (I anno – 8 crediti)
- Linguaggio C e problem solving elementare: Tecniche di Programmazione (I anno - 6 crediti)
- Problem solving avanzato, ricorsione: Algoritmi e Strutture Dati (II Anno – 8 crediti)

Esperienza passata (fino al 2019/2020)

Difficoltà nel passaggio dal primo anno (comune) al secondo anno (di indirizzo)

- Corso di Algoritmi e Programmazione fondante e DIFFICILE
- Il 60% dei nuovi frequentanti supera l'esame di Algoritmi e Programmazione nei 4 appelli - Il 40 NON lo tenta/supera

Atteggiamenti riscontrati lato studente:

- a metà corso lascio perdere tutto perchè è difficile
- mi metto in gioco, seguo le lezioni, svolgo i laboratori, studio, passo l'esame con il voto che mi merito per capacità e per applicazione.

Tecniche di programmazione

- Introdotto per rafforzare la preparazione di base ed equilibrare il carico primo/secondo anno (anticipare una parte di Algoritmi e alleggerire il secondo anno)

Esperienza passata – TP 2020/2021

- Semestre carico (ci sono altri corsi “pesanti”)
- Iscritti corsi I e II: 455 (406 ancora iscritti 21/22, 266 iscritti ASD 21/22)
- Esame superato (4 appelli):
 - 276/455 (61%), ancora iscritti 271/406 (67%), iscritti ASD 233/266 (88%)
 - Distribuzione voti:
 - 18-21: 10%,
 - 22-24: 22%,
 - 25-27: 26%,
 - 28-30L: 42%
 - => L'esame è (relativamente) facile
- Correlazione con successo in ASD?

Correlazione Esame TP e ASD (primi 2 appelli)

| TP | | ASD (prenotato) | ASD (superato) | ASD (superato/prenotato) |
|--------|-----|-----------------|----------------|--------------------------|
| NO | 33 | 2 | 0 | 0 % |
| 18-21 | 19 | 11 | 0 | 0 % |
| 22-24 | 48 | 33 | 10 | 30 % |
| 25-27 | 59 | 44 | 18 | 41 % |
| 28-30 | 107 | 84 | 57 | 68 % |
| TOTALE | 266 | 174 | 85 | 49 % |

Esame NON difficile ma voto basso indica probabili future difficoltà

Cosa serve per il corso?

Prerequisiti:

- capacità di programmare in un altro linguaggio (Python): Il C viene insegnato come SECONDO linguaggio
- Capacità logiche: comprensione del problema, identificazione di una strategia risolutiva
- Capacità pratiche: trasformare la strategia in un programma funzionante e, per quanto possibile) comprensibile, efficiente

Impegno personale continuo sul semestre:

- seguire le lezioni per capire e conoscere
- affrontare i laboratori e svolgere gli esercizi, programmare implementando i concetti capiti (richiede **tempo!**)
- non rinunciare a metà corso, non basta «svegliarsi» 2 settimane prima dell'appello

IN PRATICA...



NO: studio solo teorico senza provare a programmare

NO: limitarsi alla comprensione di programmi altrui

NO: approccio puramente «smanettone»: «non so cosa ho fatto, ma funziona»

NO: essere fuori dal contesto del corso: seguire il percorso proposto con le proprie tempistiche

IN PRATICA...



SI': mettere in pratica le nozioni apprese

SI': cimentarsi con un problema senza cercare di adattare soluzioni altrui («scopiazzare» in Rete)

SI': frequentare i laboratori e svolgere gli esercizi

SI': attenzione, concentrazione, precisione (i dettagli contano!)

SI': atteggiamento attivo/responsabile (il programma fa tutto e solo quello che il programmatore scrive, specialmente in C!)

SI': uso intelligente degli strumenti disponibili (materiale, rete, compilatore, debugger, etc.)

SI': **tempo** dedicato: studio e pratica, preparazione all'esame

DOMANDE ?