

Corso “Programmazione 1”

Capitolo 00: Presentazione del Corso

Docente: **Marco Roveri** - `marco.roveri@unitn.it`
Esercitori: **Giovanni De Toni** - `giovanni.detoni@unitn.it`
Stefano Berlato - `stefano.berlato-1@unitn.it`
C.D.L.: Informatica (INF)
A.A.: 2021-2022
Luogo: DISI, Università di Trento
URL: <https://bit.ly/2VgfYwJ>



Ultimo aggiornamento: 13 settembre 2021

Terms of Use and Copyright

USE

This material (including video recording) is intended solely for students of the University of Trento registered to the relevant course for the Academic Year 2021-2022.

SELF-STORAGE

Self-storage is permitted only for the students involved in the relevant courses of the University of Trento and only as long as they are registered students. Upon the completion of the studies or their abandonment, the material has to be deleted from all storage systems of the student.

COPYRIGHT

The copyright of all the material is held by the authors. Copying, editing, translation, storage, processing or forwarding of content in databases or other electronic media and systems without written consent of the copyright holders is forbidden. The selling of (parts) of this material is forbidden. Presentation of the material to students not involved in the course is forbidden. The unauthorised reproduction or distribution of individual content or the entire material is not permitted and is punishable by law.

The material (text, figures) in these slides is authored mostly by Roberto Sebastiani, with contributions by Marco Roveri, Alessandro Armando, Enrico Giunchiglia e Sabrina Recla.

Avviso importante: Iscrizione All'esame

Gli studenti iscritti al primo anno a INF negli AA 2021/22 e AA 2020/21.

Gli studenti iscritti al primo anno a INF o ICE nell'AA 2019/20 oppure a INF, IngCOM e IngORG nell'AA 2017/18 o 2018/19 sono tenuti ad attenersi al programma di e a sostenere l'esame con:

- Prof. RICCARDI (corso parallelo a questo) se matricola dispari,
- Prof. ROVERI (questo corso) se matricola pari.

Gli studenti iscritti al primo anno nell'AA 2016/17 o precedenti sono tenuti ad attenersi al programma di e a sostenere l'esame con:

- Prof. RICCARDI (corso parallelo a questo) se iscritti a IngCOM o IngORG
- Prof. ROVERI (questo corso) se iscritti ad INF.

Si ricorda che ESSE3 consentirà l'iscrizione all'esame secondo le regole riportate qui sopra, applicate al corso indicato nel proprio libretto elettronico. Se in dubbio, si consiglia quindi di controllare tale informazione prima di prepararsi per l'esame.

- PERIODO: I Semestre, 13/09/2021 \Rightarrow 23/12/2021
- MODALITÀ: “in presenza”¹
- DURATA: \approx 14 settimane
- Ore (accademiche) lezione: \approx 60
- Ore (accademiche) di esercitazione: \approx 40
- CREDITI: 12

¹ Salvo variazioni del caso!

Informazioni Utili: Lezione “Teoria”

- In presenza:
 - Aula T2 - Mesiano
 - Tutti i lunedì 11:30-13:30
 - Tutti i mercoledì 16:30-19:30
- On line via Zoom:
 - Topic: Programmazione 1
 - Join Zoom Meeting: <https://unitn.zoom.us/j/92225974809>
 - Meeting ID: 922 2597 4809
 - Passcode: 375001
- Per ora accesso libero, tra poco solo con email studenti.unitn.it

- In presenza:²
 - Gruppo 1 (A-P):
 - Aula B106 - Povo 2
 - Tutti i martedì 8:30-10:30
 - Tutti i giovedì 11:30-13:30
 - Gruppo 2 (Q-Z):
 - Aula B106 - Povo 2
 - Tutti i martedì 11:30-13:30
 - Tutti i giovedì 8:30-10:30
- On line via zoom (no gruppi, stesso link della teoria)
 - Tutti i martedì 8:30-10:30
 - Tutti i giovedì 8:30-10:30

²P incluso in gruppo 1, Q incluso in gruppo 2.

- **Piattaforma Moodle (materiale corso + registrazioni)**
<https://webapps.unitn.it/gestionecorsi/>
- **Esse3**
<https://www.esse3.unitn.it/>
- **Sito Web del corso**
<https://bit.ly/2VgfYwJ>

Ricordatevi di iscrivervi al corso sulla piattaforma Moodle!

- Insegnare i fondamenti della programmazione imperativa.
- Come linguaggio, viene usato (un sottinsieme stretto del) C++, visto prettamente come linguaggio di programmazione imperativo
 - \implies in questo corso aspetti di programmazione Object-Oriented non verranno analizzati (o verranno analizzati solo marginalmente).

Coding is not Programming

Coding

- I computer “capiscono” Codice Macchina (si veda il corso di Calcolatori), di difficile comprensione dagli esseri umani:
 - Linguaggi di Programmazione.
- Coding è il processo di usare un linguaggio di programmazione per fare sì che un computer faccia quello che lo sviluppatore vuole.
- In parole semplici, coding consiste in scrivere molteplici linee di codice per creare un programma software (application, game, or website).

Programming

- Sebbene coding sia una fase molto importante del ciclo di sviluppo software, non è una componente essenziale!
- Per creare una applicazione, molteplici passi sono necessari: planning, design, testing, deployment, e maintenance.
- Programming è una attività complessa che comprende non solo coding, ma anche altre attività e.g. **progettare, analizzare e implementare** algoritmi, **capire** le strutture dati, e **risolvere problemi**.
- Programming è essenziale. Per scrivere codice è necessario essere trasparenti con lo schema o la struttura del programma: e.g. scrivere pseudocodice per descrivere la logica di un algoritmo, ed usarlo come strumento per spiegare l'algoritmo ad un programmatore (coder).

<https://hackr.io/blog/coding-vs-programming-difference-you-should-know>

Coding is not Programming

Coding

Scopo È un processo di conversione di un insieme di istruzioni in un linguaggio comprensibile ad un computer

Skill Come coder, è richiesta buona conoscenza della sintassi e semantica del linguaggio di programmazione scelto

<https://hackr.io/blog/coding-vs-programming-difference-you-should-know>

Programming

Oltre al coding, comprende la definizione dei requisiti, problem solving, scrivere pseudocodice, pensare algoritmi, ottimizzare, testing, creare eseguibili, manutenzione

Come programmatore, sono richieste oltre alle coding skill, capacità di pensiero (high-level thinking) e di analisi

If Programming is the process of writing a whole book.

Then Coding is just about writing a single chapter of the book.

Pre-requisiti

- Nessuna conoscenza informatica pregressa
- Nozioni di base di matematica
- Padronanza della lingua italiana, scritta e orale
- Discrete capacità logiche/analitiche

Programma di massima (non necessariamente in ordine)

- Concetti generali
- Sviluppo di un programma
- I/O standard, I/O su files (argc & argv)
- Variabili e costanti
- Tipi (booleani, interi, reali, caratteri)
- Sintassi del C++ (cenni)
- Istruzioni elementari
- Istruzioni strutturate (condizioni, cicli)
- Funzioni e passaggi di parametri
- Array, Stringhe
- Organizzazione di un programma su più file: scope, visibilità e durata

Programma di massima (2)

- Puntatori e Riferimenti, Algebra dei puntatori
- Array e puntatori, Passaggio di parametri per puntatore
- Struct
- Allocazione dinamica di memoria
- Allocazione dinamica di array, struct
- [Cenni di programmazione Object-Oriented]
- Strutture dati fondamentali (liste, stack, code) e loro realizzazione tramite array e struct
- Strutture dati dinamiche e loro implementazione con liste concatenate
- Alberi binari

Riferimenti

- Appunti delle lezioni
- Handouts/slides “Corso Programmazione 1” (disponibile su Moodle/sito)
- **ESEMPI / Materiale didattico aggiuntivo** (disponibile su Moodle/sito)
- Libro di testo suggerito (alternativi):
 - John R. Hubbard. “Programming with C++” (2ed)
McGraw Hill (ENGL): ISBN: 0-07-135346-1,
<http://www.mathcs.richmond.edu/~hubbard/Books.html>
(ITA): ISBN: 88-386-5045-4 “Programmare in C++” (non più disponibile nuovo)
 - Luis Joyanes Aguilar “Fondamenti di programmazione in C++” McGraw Hill, ISBN: 9788838664779
http://www.catalogo.mcgraw-hill.it/catLibro.asp?item_id=2299
- Riferimento per il linguaggio C++:
 - Bjarne Stroustrup “Il Linguaggio C++” Pearson, ISBN: ISBN 9788871920788 (4^a ed.)
http://www.pearson.it/opera/pearson/0-2613-c_%2B%2B_linguaggio_libreria_standard_principi_di_programmazione

- Prova pratica di programmazione al calcolatore
 - 5 appelli: 2 gennaio/febbraio, 2 giugno/luglio, 1 settembre
 - Durata: 2 ore
 - 3 esercizi + uno facoltativo molto difficile (per la lode)
 - La modalità di svolgimento dipenderà da come evolve la situazione.
 - o svolti in laboratorio (aule A201/202 o B106), sotto linux Ubuntu
 - o svolti remotamente con meccanismo da definirsi
 - Ammesso l'uso di editor/compilatore/debugger, nessun altro strumento, nessun testo
 - ...

Suggerimenti

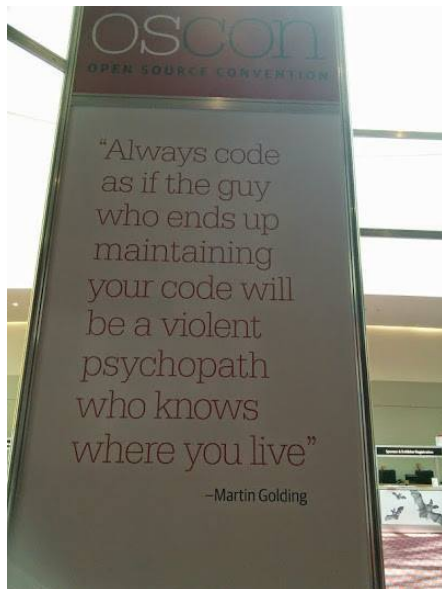
- FREQUENTARE/SEGUIRE LE LEZIONI E LE ESERCITAZIONI!!!!
 - N.B.: ogni cosa detta a lezione è potenziale argomento di esame, non solo ciò che è esplicitamente contenuto nelle slide/materiale
- Chiedere ciò che non si capisce
- Non posticipare lo studio:
 - Studiare/provare a implementare dopo ogni lezione
 - ⇒ molto più efficiente per massimizzare il risultato a parità di sforzo
- Svolgere **sempre** gli “esercizi proposti”
- Implementare, implementare, implementare!
 - N.B.: la capacità di capire i messaggi del compilatore e/o del debugger è parte del bagaglio di conoscenze implicitamente richiesto per passare l'esame.
- ...

Suggerimenti (2)

- ...
- Quando si implementa:
 - Pensare e farsi uno schema prima si cominciare a scrivere codice
 - **Adottare sempre le soluzioni più semplici possibile**
 - **Scrivere il codice in modo il più chiaro e leggibile possibile**
 - Indentare il codice!!!
 - Lasciare spazi (ma non troppi spazi)
 - Usare nomi significativi e/o convenzionali
 - ...
 - **Seguire “coding standards”:**
 - Un “coding standard” da un aspetto “uniforme” a codice scritto da diversi programmatori, migliora la leggibilità, la manutenibilità del codice, contribuisce a ridurre la complessità di comprensione, favorisce il riuso del codice, e contribuisce positivamente all’identificazione di errori.

“Dovete programmare sempre come se il professore che deve correggere il vostro codice fosse uno psicopatico violento che sa dove abitate.”

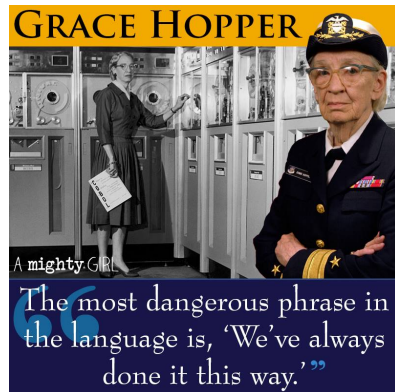
[Marco Roveri and Roberto Sebastiani]



Vedere anche: <http://www.journaldev.com/240/my-25-favorite-programming-quotes-that-are-funny-too>

Suggerimenti (3)

- Suggerimento generale: **affrontare la programmazione con umiltà**.
 - Ultime parole famose:
"Tanto io sono già capace di programmare"
- Se necessario, essere pronti a **cambiare il modo di lavorare**
 - Ultime parole famose #2:
"Ho sempre fatto in questo modo"



Si vedano i risultati degli esami anni precedenti:

<https://esamionline.unitn.it/course/view.php?id=1873>

Interazione con docente ed esercitatori

- Domande durante la lezione
 - fortemente incoraggiate
- Domande durante l'intervallo
 - piena disponibilità
- Ricevimento: su appuntamento, con orario da concordare di volta in volta (per email o a lezione),
 - Solo durante il periodo delle lezioni
- Invio di email
 - con moderazione e entro certi limiti (vedi slide successive)
- MAI PER TELEFONO!

Nota importante

È attivo un servizio di **tutorato** per studenti con difficoltà o lacune pregresse, gestito da studenti degli ultimi anni o dottorandi.

Usi ed abusi dell'email (1)

Ogni email va inviata da `nome.cognome@studenti.unitn.it`, e deve sempre avere in copia tutti gli esercitatori!

Buoni motivi per inviarci una email

- Richieste di ricevimento
- Segnalazioni di eventuali refusi o errori nelle slides/programmi, problemi di accesso al sito web, ecc.
- Segnalazioni di problemi oggettivi (sovrapposizioni di orario,...)
- Segnalazione di problemi individuali particolari, ad es:
 - studenti diversamente abili o con disturbi specifici dell'apprendimento (DSA) certificati
 - studenti non ancora iscritti/in corso di trasferimento
 - studenti lavoratori
 - studenti con problemi o situazioni particolari(In questi casi il colloquio diretto è comunque preferibile.)
- ...

Usi ed abusi dell'email (2)

Motivi inutili per inviarci una email

- Giustificazioni di assenze/risultati ecc.
 - “Non sono potuto venire a lezione perche'...”
 - “Le prossime due settimane saró assente...”
 - “Ho sbagliato il terzo esercizio dell'esame, ...”
 - “Mi sono iscritto all'appello ma non sono potuto venire ...”
 - ...

Usi ed abusi dell'email (3)

Cattivi motivi per inviarci una email

- Richieste di informazioni già date a lezione o presenti sul sito
- Richieste di risoluzione di problemi individuali
 - *“Perche’ il programma in allegato non funziona?”*
 - *“Ho istallato la versione ZBX1002.34.56 del programma/sistema operativo XYZ, ma mi riporta i seguenti errori:...”*
 - *“Vorrei comprare un pc/laptop/tablet/etc., va meglio il modello ZK45.5 della XYZ o il KW32.7 della ZYX?”*
- Domande/richieste/affermazioni sull'esame
 - *“Sono usciti i risultati?”*
 - *“Perche’ non sono ancora usciti i risultati?”*
 - *“io credo di aver fatto questo errore, ma è stata una distrazione, ...”*
 - *“io avevo fatto tutto giusto, poi il computer mi ha cancellato il file...”*
 - *“lo giuro, in aula compilava!”*
 - ...

Usi ed abusi dell'email (4)

Infine, la buona educazione e il rispetto nei confronti del docente e degli esercitatori via email (come nell'interazione diretta) è d'obbligo.

(Vedi anche: “*Bad Email Reply - What not to say to your professor...*”

<https://www.youtube.com/watch?v=zSNc8F9tqzY>.)

PS:

Per evitare gaffes o situazioni sgradevoli, ricordatevi che anche i vostri docenti/esercitatori usano internet, social media, ecc:

<https://www.facebook.com/spottedunitn/posts/410484665765366?fref=nf&pnref=story>

[cfr Sebastiani]