



UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO DI
MATEMATICA E INFORMATICA

Corso di Programmazione II

Alessandro Ortis

Image Processing Lab - iplab.dmi.unict.it

alessandro.ortis@unict.it

www.dmi.unict.it/ortis/





Info&contacts

Alessandro Ortis, *PhD*

Email: alessandro.ortis@unict.it

Personal webpage: <http://www.dmi.unict.it/ortis/>



Università
di Catania



IPlab@UNICT



<http://iplab.dmi.unict.it>

Core Competences: **Computer Vision and Multimedia**

Overall Team (~20 – Lead by Prof. S. Battiato)

- **Results:** >30 patents, >400 papers
- **Applications to:**
 - Multimedia Forensics and Security
 - EgoCentric Vision
 - Cultural Heritage
 - Social Media Mining
- **R&D projects:**
 - STM (Catania, Grenoble): Joint R&D lab (since 2007), Ph.D. (>10), grants (>10)
 - Funded projects: ENIAC (1), PO/FESR (4), MISE HZ2020(1), other (4)
- **International Events:** ICVSS (since 2007), MISS (since 2014) ICIAP 2017, ACIVS 2015, VISAPP

4 FULL PROFESSORS
4 ASSISTANT PROFESSORS
4 RESEARCH FELLOWS
4 POST DOCS
>20 PHD STUDENTS



Università
di Catania

Workshop on Generative AI and Computer Vision

Aula Magna Dipartimento di Matematica e Informatica

Invited Speakers



Prof. Daniele Ravi
Professore Associato
University of Hertfordshire



Prof. Sebastiano Battiato
Professore Ordinario
Università di Catania



Ricercatori IPLab
Università di Catania
Dipartimento di Matematica e
Informatica

Programma

- 09.00 Presentazione Gruppo di Ricerca
Prof. S. Battiato
- 09.30 Pillole di Ricerca
A. Ortis, G. Fargetta, L. Guarnera, F. Guarnera, M.O. Spata
- 10.30 Invited Speaker
Prof. Daniele Ravi - University of Hertfordshire
- 11.30 Coffee Break
- 12.00 Incontro con le Aziende
- 12.30 Progetti di Ricerca e Tesi

Contatti

sebastiano.battiato@unict.it **Prof. S. Battiato**
georgia.fargetta@unict.it **Dott.ssa G. Fargetta**

<https://iplab.dmi.unict.it>

19 Marzo 2024

Luogo	Ora
Aula Magna DMI	9.00 A.M.

Aziende












I nostri strumenti

Strumenti essenziali:

- Compilatore C++ (gcc, MinGW, ...)
- Editor di testo

Opzionalmente:

- IDE (es., Visual Studio Code, Codeblocks, DevC++, etc.)

Possedere un laptop è altamente consigliato

Compilatore/ambiente C++

DevC++ (Windows)

<http://www.bloodshed.net/devcpp.html>

IDE (Integrated Development Environment) per programmare in C/C++.

- Include il porting Mingw del compilatore GCC (GNU Compiler Collection).
- Rilasciato sotto licenza GPL (GNU General Public License).

Compilatore/ambiente C++

Alcune note per l'uso di DevC++:

Selezionare, attraverso il menu a tendina, la versione appropriata del compilatore da usare (32 bit o 64 bit).

Combinazione di tasti CTRL+n per lanciare nuova finestra di editing per singolo file sorgente.

Salvare il file sorgente come <miofile>.cpp, ES: pippo.cpp

Compilazione: F9, Esecuzione del programma: F10

Compila ed esegui in successione: F11

Compilatore/ambiente C++

Code::Blocks (Windows, Linux, Apple)

IDE (Integrated Development Environment) per programmare in C/C++ (anche Fortran): <http://www.codeblocks.org>

- Multiplatforma (Linux, Windows, Mac OS)
- Windows: Code::blocks include il porting Mingw del compilatore GCC (GNU Compiler Collection).
- Licenza GPL (GNU General Public License).

Compilatore/ambiente C++

È consigliato l'uso di un semplice editor di testo e la compilazione mediante riga di comando.

Esempio di editor: SciTE <https://www.scintilla.org/SciTE.html>

- Interfaccia semplice, minimale.
- Syntax highlighting per molti linguaggi, anche C e il C++.
- Dal menu Tool è possibile invocare il compilatore gcc, eseguire successivamente il programma (quindi analogo al tasto F11 di Dev-C++) e quindi e output in un frame dell'applicazione

Esempio di compilazione/esecuzione:

```
>> g++ pippo.cpp -o mio_programma  
>> ./mioprogramma
```

Da C a C++ (FAQ)

È facile imparare C++ conoscendo C?

Assolutamente sì! Un primo passo è programmare in C++ come se si stesse programmando in C, notando le eventuali differenze. Ad esempio, il C++ fornisce un controllo dei tipi più safe.

Da C a C++ (FAQ)

È facile imparare C++ conoscendo C?

Assolutamente sì! Un primo passo è programmare in C++ come se si stesse programmando in C, notando le eventuali differenze. Ad esempio, il C++ fornisce un controllo dei tipi più safe.

Qual è la differenza tra C e C++?

C++ è un discendente diretto di C, con un controllo sui tipi più forte ed un più ampio spettro di paradigmi di programmazione disponibili (astrazione, OOP, programmazione generica).

Da C a C++ (FAQ)

È facile imparare C++ conoscendo C?

Assolutamente sì! Un primo passo è programmare in C++ come se si stesse programmando in C, notando le eventuali differenze. Ad esempio, il C++ fornisce un controllo dei tipi più safe.

Qual è la differenza tra C e C++?

C++ è un discendente diretto di C, con un controllo sui tipi più forte ed un più ampio spettro di paradigmi di programmazione disponibili (astrazione, OOP, programmazione generica).

Possiamo dire che C è un sottoinsieme di C++?

Non in senso matematico (quindi no!). Infatti ci sono codici validi in C che non sono validi in C++ oppure danno output diversi (vi sfido a trovarli!).

Suggerimenti: utilizzo di void*, cast implicito, utilizzo delle struct...

```
struct mystruct {};  
  
int main(){  
    struct mystruct x;  
    printf("%d", (int)sizeof(x));  
    return 0;  
}
```

Obiettivi Formativi

- Acquisire **metodi** di programmazione
 - Imparare a ragionare da informatici
 - Paradigma OOP (Object Oriented Programming)
- Acquisire e sviluppare capacità di
 - Comprendere le proprietà fondamentali di diversi algoritmi e strutture dati per essi
 - Implementare gli algoritmi studiati in C++ in modo da ottenere soluzioni affidabili ed efficienti

Il Syllabus di ciascun corso si trova online:

<https://web.dmi.unict.it/corsi/l-31/programmi>

Perché “Efficienti”?

- I computer diventano sempre più rapidi e potenti perché curarsi di scrivere programmi efficienti?
- Perché l'efficienza è più importante della potenza

Tempo n^5 (1 operaz. al secondo)

Dimensione dell'input (n)	2	5	10	20	30	40	70
Tempo di calcolo	32	52	28	37	9	39	54
	Secondi	minuti	ore	giorni	mesi	mesi	anni

- Portiamo a 10 op/sec la potenza di calcolo

Dimensione dell'input (n)	2	5	10	20	30	40	70
Tempo di calcolo	3,2	5,2	2,8	3,7	28	3,9	5,4
	Secondi	minuti	ore	giorni	giorni	mesi	anni

Tempo n^2 (1 operaz. al secondo)

Dimensione dell'input (n)	2	5	10	20	30	40	70
Tempo di calcolo	4 sec	25 sec	100 sec	7 min	15 min	27 min	82 min

- Meglio del metodo precedente implementato su una macchina 10 volte più veloce!

Dimensione dell'input (n)	2	5	10	20	30	40	70
Tempo di calcolo	3,2 Secondi	5,2 minuti	2,8 ore	3,7 giorni	28 giorni	3,9 mesi	5,4 anni

Attività formative

- 72 ore di lezione frontale (teoria+pratica)
- L'acquisizione dei metodi e delle competenze professionali della materia è sostenuta da
 - Frequenza delle lezioni (consigliata)
 - Studio individuale (tanto)
 - Studio del testo
 - Partecipazione al tutorato (?)

Tutorato 2023-2024

08/02/24	3	15:00-17:00
15/03/24	3	15:00-17:00
22/03/24	3	15:00-17:00
29/03/24	3	15:00-17:00
05/04/24	3	15:00-17:00
12/04/24	3	15:00-17:00
19/04/24	3	15:00-17:00
26/04/24	3	15:00-17:00
03/05/24	3	15:00-17:00
10/05/24	3	15:00-17:00
17/05/24	3	15:00-17:00
24/05/24	3	15:00-17:00
31/05/24	3	15:00-17:00
07/06/24	3	15:00-17:00
14/06/24	3	15:00-17:00

Attività formative

- 72 ore di lezione frontale (teoria+pratica)
- L'acquisizione dei metodi e delle competenze professionali della materia è sostenuta da
 - Frequenza delle lezioni (consigliata)
 - Studio individuale (tanto)
 - Studio del testo
 - Partecipazione al tutorato (?)

Attività formative

- 72 ore di lezione frontale (teoria+pratica)
- L'acquisizione dei metodi e delle competenze professionali della materia è sostenuta da
 - Frequenza delle lezioni (consigliata)
 - Studio individuale (tanto)
 - Studio del testo
 - Partecipazione al tutorato (?)

Attività formative

- 72 ore di lezione frontale (teoria+pratica)
- L'acquisizione dei metodi e delle competenze professionali della materia è sostenuta da
 - Frequenza delle lezioni (consigliata)
 - **Studio individuale (tanto)**
 - Studio del testo
 - Partecipazione al tutorato (?)

Attività formative

- 72 ore di lezione frontale (teoria+pratica)
- L'acquisizione dei metodi e delle competenze professionali della materia è sostenuta da
 - Frequenza delle lezioni (consigliata)
 - **Studio individuale (tanto)**
 - Studio del testo
 - Partecipazione al tutorato (?)

Attività formative

- 72 ore di lezione frontale (teoria+pratica)
- Acquisizione dei metodi e delle competenze professionali della materia è sostenuta dalla frequenza delle lezioni (consigliata)

– **Studio individuale (tanto)**

- Studio del testo
- Partecipazione al tutorato (?)

Testi

Luis Joyanes Aguilar

Fondamenti di Programmazione in C++

McGraw-Hill

T. H Cormen

Introduction to Algorithms

The MIT Press

Consigliati:

C++ Tecniche avanzate di programmazione Apogeo

Algoritmi in C++ (terza edizione) Pearson Education Italia

Effective C++ e More Effective C++ Addison-Wesley

Altri riferimenti utili

- <https://cppreference.com>
- <https://cplusplus.com>
- <https://www.hackerrank.com/domains/cpp>

Relazione con altri insegnamenti

Propedeuticità richieste: per sostenere l'esame di Programmazione 2, dovrete aver **già superato**:

- Programmazione 1
- Strutture Discrete

Programma delle lezioni - I

Parte 1: Programmazione ad oggetti

- Classi e Oggetti
- Classi derivate
- Templates
- Sovraccaricamento degli operatori

Programma delle lezioni - II

Parte 2: Cenni di complessità e tecniche ricorsive

- Cenni di complessità asintotica
- Ricorsione ed algoritmi di ordinamento basati sulla ricorsione

Programma delle lezioni - III

Parte 3: Strutture Dati

- Liste
- Pile e code
- Alberi
- Grafi

Parte 1, 2 e 3:

- Tantissimi esercizi

Kahoot!



<https://kahoot.com/>

Modalità d'esame



Tipicamente le tre prove avvengono nell'arco della stessa giornata, se il numero di prenotati lo permette, per evitare sovrapposizione con gli appelli di altre materie del primo anno.

La prenotazione è obbligatoria.

Modalità d'esame



1. Prova scritta:

- Durata 1h
- 30 quesiti a risposta multipla
- Tutti i fogli utilizzati vanno consegnati

Modalità d'esame



2. Prova pratica di laboratorio:

- Realizzazione di un codice in C++
- Durata: circa 2 ore
- Compilazione mediante g++ su una macchina generica
- Sarà possibile consultare la documentazione di C++ durante la prova (no Internet!)

Modalità d'esame



3. Prova Orale:

- Compilazione ed esecuzione del progetto
- Quesiti sull'implementazione
- Quesiti su tutto il programma didattico

Il voto finale è il risultato di una valutazione complessiva delle tre prove.

Avvisi e comunicazioni

- Pagina degli avvisi del Corso di Laurea in Informatica: [LINK](#)
- Pagina degli avvisi dei DOCENTI del Corso di Laurea in Informatica: [LINK](#)

Domande?