



UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Informatica

Modulo di Programmazione

INFORMATICA
MODULO DI PROGRAMMAZIONE
INTRODUZIONE

mauro.pelucchi@gmail.com

Mauro Pelucchi

2023/2024

Programm

~ 2 ore
Getting Started

- 1) C++ Intro
- 2) Getting Started
- 3) Debugging Tools
- 4) Control-Flow Constructs (if, switch)

~ 2 ore
C++ Basics

- 1) Operators and Expressions
- 2) Basic I/O
- 3) Control-Flow Constructs (while, for)

~ 2 ore
C++ Basics

- 1) Small Applications

~ 2 ore
Pointers &
Vectors

- 1) Pointers
- 2) Vectors

Programma

~ 2 ore
Functions in C++

- 1) Functions
- 2) Parameters Passing

~ 2 ore
Complex
Structure

- 1) Struct
- 2) List & Queue

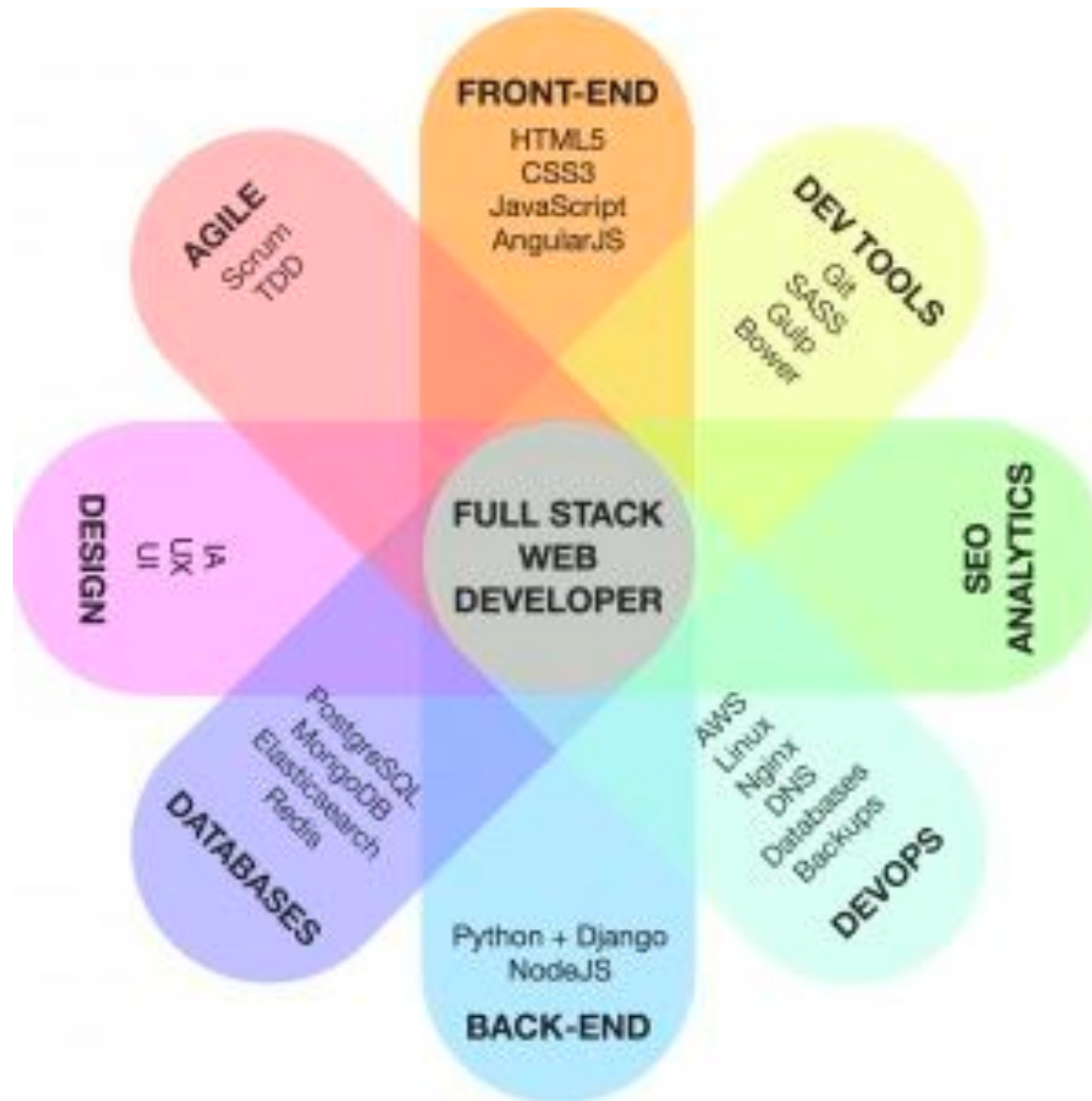
~ 2 ore
Applications

- 1) File I/O
- 2) Sorting

~ 2 ore
Applications

- 1) Small Applications

Full stack software engineer



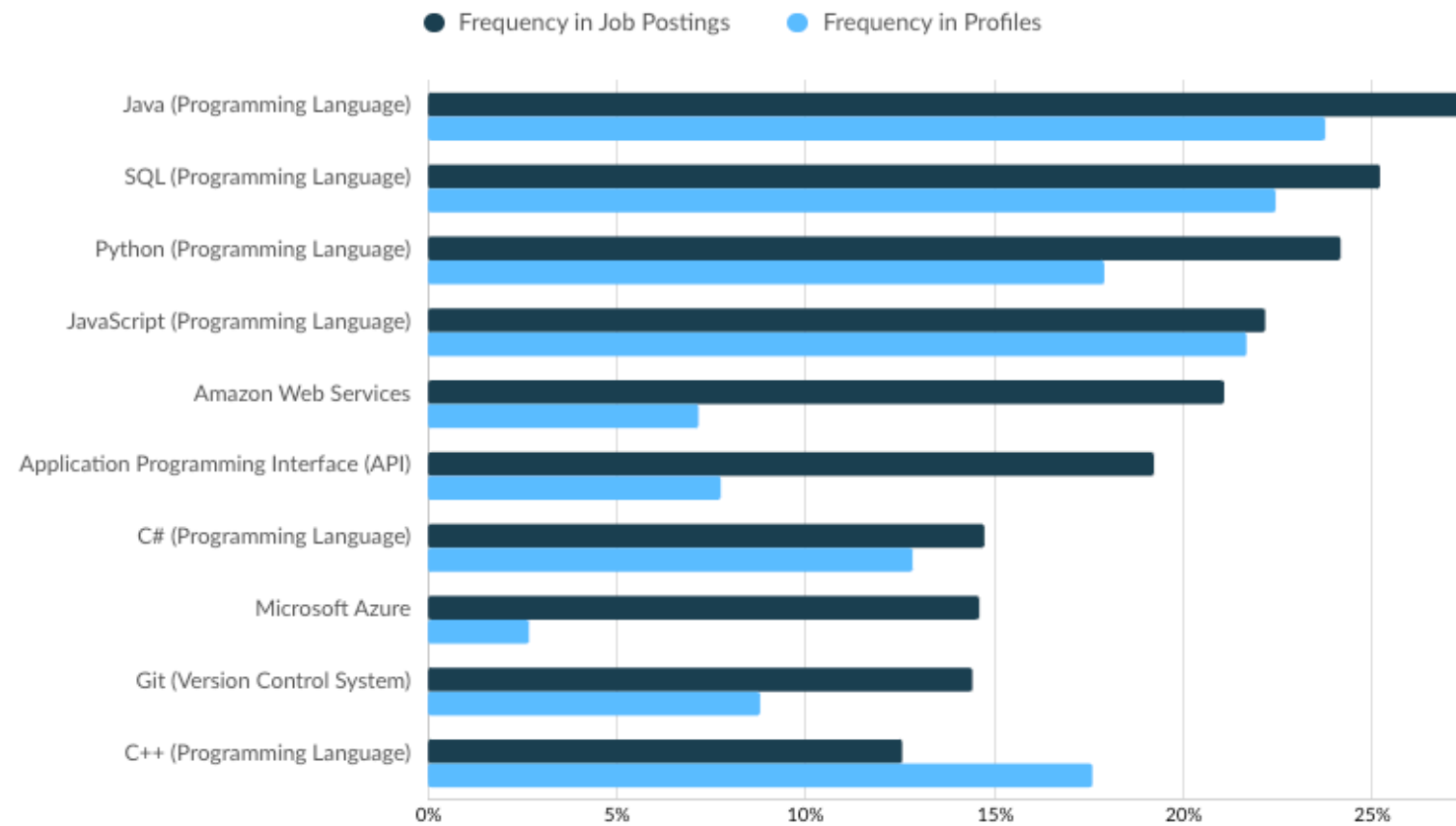
Modulo di programmazione

Perché?











Perché C++?

Software Developer / Engineer

Top Software Skills



Skills	Postings	% of Total Postings	Profiles	% of Total Profiles
Java (Programming Language)	174,950	28%	478,382	24%
SQL (Programming Language)	159,280	25%	451,653	22%
Python (Programming Language)	152,700	24%	360,601	18%
JavaScript (Programming Language)	139,978	22%	436,131	22%
Amazon Web Services	133,234	21%	143,803	7%
Application Programming Interface (API)	121,454	19%	156,329	8%
C# (Programming Language)	93,095	15%	258,564	13%
Microsoft Azure	92,257	15%	54,132	3%
Git (Version Control System)	91,014	14%	176,794	9%
C++ (Programming Language)	79,227	13%	354,103	18%

Sep 2023	Sep 2022	Change	Programming Language		Ratings	Change
1	1			Python	14.16%	-1.58%
2	2			C	11.27%	-2.70%
3	4	^		C++	10.65%	+0.90%
4	3	v		Java	9.49%	-2.23%
5	5			C#	7.31%	+2.42%
6	7	^		JavaScript	3.30%	+0.48%
7	6	v		Visual Basic	2.22%	-2.18%
8	10	^		PHP	1.55%	-0.13%
9	8	v		Assembly language	1.53%	-0.96%
10	9	v		SQL	1.44%	-0.57%

<https://www.tiobe.com/tiobe-index/>

TIOBE Index for C

Source: www.tiobe.com



TIOBE Index for C++

Source: www.tiobe.com



<https://www.tiobe.com/tiobe-index/>

C

- created by Dennis Ritchie, AT&T Bell Labs in 1970s
- international standard ISO/IEC 9899:2011 (informally known as “C11”)
- available on wide range of platforms, from microcontrollers to supercomputers; very few platforms for which C compiler not available
- procedural, provides language constructs that map efficiently to machine instructions
- does not directly support object-oriented or generic programming
- application domains: system software, device drivers, embedded applications, application software
- greatly influenced development of C++
- when something lasts in computer industry for more than 40 years (outliving its creator), must be good

C++

- created by Bjarne Stroustrup, Bell Labs
- originally C with Classes, renamed as C++ in 1983 most recent specification of language in ISO/IEC 14882:2014 (informally known as “C++14”)
- procedural
- loosely speaking is superset of C
- directly supports object-oriented and generic programming
- maintains efficiency of C
- application domains: systems software, application software, device drivers, embedded software, high-performance server and client applications, entertainment software such as video games, native code for Android applications
- greatly influenced development of C# and Java

Java

- developed in 1990s by James Gosling at Sun Microsystems (later bought by Oracle Corporation)
- de facto standard but not international standard
- usually less efficient than C and C++
- simplified memory management (with garbage collection)
- direct support for object-oriented programming
- application domains: web applications, Android applications

- developed by Microsoft, team led by Anders Hejlsberg
- most recent language specifications not standardized by ECMA or ISO/IEC
- intellectual property concerns over Microsoft patents
- object oriented

Objective C

- developed by Tom Love and Brad Cox of Stepstone (later bought by NeXT and subsequently Apple)
- used primarily on Apple Mac OS X and iOS
- strict superset of C
- no official standard that describes Objective C
- authoritative manual on Objective-C 2.0 available from Apple

Why Learn C++?

General Purpose

Powerful yet
efficient

Loosely
speaking,
includes C as
subset;

Memory
Management

Object-Oriented

Easy to move
from C++ to
other languages

Materiale e Lezioni

Materiale

- Slide e materiale
- <https://tinyurl.com/ModProg2023>
- GitHub
- https://github.com/mauropelucchi/unibg_dev_courses_2023

Come seguire le lezioni?

- Il materiale verrà caricato sulla cartella condivisa GDrive
- Codice sorgente e altro materiale verrà reso disponibile attraverso GitHub
- Software/Tools necessari sul proprio notebook:
 - DevCpp oppure Visual Studio Code <https://code.visualstudio.com/>
 - Account GitHub (consigliato)
 - https://www.onlinegdb.com/online_c++_compiler
- Altro materiale verrà comunicato durante il corso

Come seguire le lezioni?

- Lezioni in Remoto
 - Canale Microsoft Teams
 - Team A / Team B
 - **Giovedì dalle 8:30 alle 10:30**
 - **Giovedì dalle 10:30 alle 12:30**
- **Ricevimento studenti —> mandare una mail oppure via chat via teams e fissare un appuntamento!!! Caldamente consigliato per qualsiasi dubbio!!!**
- Email
 - mauro.pelucchi@unimib.it
 - mauro.pelucchi@gmail.com
- Registrazione delle lezioni disponibili su Teams

Ricevimento studenti

- NON c'è orario di ricevimento per le esercitazioni
- MA il modo migliore (e più sicuro) per essere ricevuti e ascoltati è fissare un appuntamento (via e-mail)
- Utilizzare la mail dell'università XX@studenti.unibg.it
 - Preferibilmente, non utilizzare indirizzi e-mail
 - come **masterOfWarCraft@isp.com** (non è professionale !)
 - Scrivere in modo sintetico ma chiaro
 - Identificandosi precisamente [nome, matricola, corso di laurea]
 - Spiegare brevemente lo scopo dell'incontro:
 - chiarimenti didattici su ...
 - Dare all'e-mail un **Oggetto** sintetico ma preciso:
 - Es: "INFORMATICA MODULO Programmazione..."

Svolgimento della lezione

- Sezione teoria + Dev session [40 minuti]
 - Degli esercizi proposti soltanto alcuni verranno risolti subito per ricapitolare i concetti su cui si focalizzerà l'esercitazione oppure per introdurre qualche caratteristica del linguaggio.
- Assignment [30 min]
 - Almeno un esercizio, tipicamente due, saranno svolti in autonomia da ciascuno studente, o coppia di studenti.
 - Il tempo previsto per ciascun esercizio è stimato in circa 20-25 minuti.
- Presentazione del vostro lavoro e commento [20 minuti]
- Q&A
- Le soluzioni degli esercizi saranno disponibili solo al termine dell'esercitazione sul repository GitHub del corso.
 - Ogni settimana (circa), come materiale complementare, verranno pubblicate tracce di problemi (Homeworks) senza soluzione, che si invita a risolvere e commentare assieme al docente.

Svolgimento della lezione

- Si raccomanda vivamente di NON rimandare l'occasione di fare pratica ai soli momenti di studio individuale ma di usare la sessione di laboratorio per chiarire i propri dubbi
- Sulla base delle statistiche fatte negli anni precedenti si raccomanda anche a coloro che ritengono di avere una certa padronanza della programmazione di NON sottovalutare il corso.