



## Introduzione



**Anno accademico 2018/19**  
Ingegneria del Software

Tullio Vardanega, [tullio.vardanega@math.unipd.it](mailto:tullio.vardanega@math.unipd.it)

Corso di Laurea in Informatica, Università di Padova

1/40



Introduzione

## Cosa facciamo in questo corso – 1

- ❑ **Apprendiamo un metodo di lavoro necessario per la professione informatica**
  - **Gestire il tempo**
    - Disponibilità, scadenze, conflitti, priorità
  - **Collaborare**
    - Fissare obiettivi, dividersi compiti, verificare progressi, riportare difficoltà
  - **Assumersi responsabilità**
  - **Auto-apprendere**
    - "Imparare a imparare" è una essenziale competenza trasversale
- ❑ **Integriamo la teoria con la pratica**

Common soft skills

- Strong work ethic
- Positive attitude
- Good communication skills
- Time management abilities
- **Problem-solving skills**
- Acting as a team player
- Self-confidence
- Ability to accept and learn from criticism
- Flexibility/adaptability
- Working well under pressure



Corso di Laurea in Informatica, Università di Padova

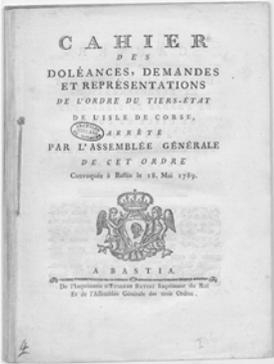
3/40



Introduzione

## Impariamo a conoscerci ...





Corso di Laurea in Informatica, Università di Padova

2/40



Introduzione

## Come vogliamo imparare

- ❑ **La conoscenza passa dalla comprensione profonda dei significati**
- ❑ **Vogliamo costruirci un glossario**
  - Raccolta di termini/concetti bisognosi di spiegazione
  - Registrati in un ordine che ne faciliti la localizzazione
  - Corredati dalla specifica del loro significato e ogni altra informazione utile a comprenderli
- ❑ **Consolidando la teoria con la pratica**

Cominciando dai termini evidenziati in colore nelle diapositive



Corso di Laurea in Informatica, Università di Padova

4/40

 **Introduzione**

## Come lo facciamo – 1

- ❑ **Tramite una attività annuale**
  - Estesa su due semestri
- ❑ **Con impegni sia collaborativi che individuali**
  - Un progetto didattico ✓
  - Una prova scritta individuale
- ❑ **Relazionandoci con terze parti** ✓
- ❑ **Usando tecnologie innovative** ✓

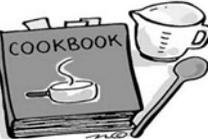
Sfide nuove e impegnative

Corso di Laurea in Informatica, Università di Padova 5/40

 **Introduzione**

## Cosa è un progetto

- ❑ **Quali attività comprende?**
  - **Pianificazione**
    - Gestire risorse (persone, tempo, fondi, strumenti) e responsabilità
  - **Analisi dei requisiti**
    - Definire cosa bisogna fare
  - **Progettazione (*design* ≠ *project*)**
    - Definire come farlo
  - **Realizzazione**
    - Farlo quel che serve, perseguendo qualità
    - Verificando che quanto prodotto non contenga errori
    - Validando i risultati rispetto alle attese



Corso di Laurea in Informatica, Università di Padova 7/40

 **Introduzione**

## Glossario – 1

- ❑ **Progetto**
  - **Insieme di attività e compiti che**
    - Devono raggiungere determinati obiettivi con specifiche fissate
    - Hanno date d'inizio e di fine fissate
    - Possono contare su limitate disponibilità di risorse (persone, tempo, denaro, strumenti)
    - Consumano tali risorse nel loro svolgersi

Fonte: Harold Kerzner (1940-), uno dei maggiori esperti mondiali di *project management*

Corso di Laurea in Informatica, Università di Padova 6/40

 **Introduzione**

## Cosa non è un progetto?

- ❑ **Molti progetti SW hanno esito deludente**
  - Per difficoltà di avvio o di svolgimento
  - Per cambi di tecnologia necessari ma imprevisti
  - Per difetti residui nel prodotto finale
- ❑ **A volte falliscono completamente**
  - Perché diventano vecchi prima della fine
  - Perché gli incaricati sono incapaci
  - Perché bruciano le risorse assegnate



Corso di Laurea in Informatica, Università di Padova 8/40



Introduzione

## Cosa non è un progetto – 2

- ❑ *One is blinded to the fundamental uselessness of their products, by the sense of achievement one feels in getting them to work at all*
- ❑ *In other words, their fundamental design flaws are completely hidden by their superficial design flaws*

Fonte: Douglas Adams, "The Hitchhikers Guide to the Galaxy", 1979

Corso di Laurea in Informatica, Università di Padova
9/40



Introduzione

## Glossario – 2

- ❑ **Software engineering [SWE]**
  - Disciplina per la realizzazione di prodotti SW così impegnativi da richiedere il dispiego di attività collaborative
  - Capacità di produrre "in grande" e "in piccolo"
  - Garantendo qualità (efficacia)
  - Contenendo costi e tempi di produzione (efficienza)
  - Lungo l'intero ciclo di vita dello sviluppo del prodotto richiesto

Corso di Laurea in Informatica, Università di Padova
11/40



Introduzione

## Cos'è l'ingegneria (*engineering*)

*Engineering: application of scientific and mathematical principles to practical ends*  
 Fonte: American Heritage Dictionary

- ❑ **Applicazione (e non creazione!) di principi noti e autorevoli → *best practice***
  - Non esplorare ma applicare
- ❑ **"Practical ends" che spesso sono civili e sociali**
  - Infrastrutture e servizi, non solo prodotti di consumo
- ❑ **Questa caratteristica comporta responsabilità etiche e professionali**

Corso di Laurea in Informatica, Università di Padova
10/40



Introduzione

## Glossario – 3



- ❑ **Efficacia**
  - Misura della capacità di raggiungere l'obiettivo prefissato
- ❑ **Efficienza**
  - Misura dell'abilità di raggiungere l'obiettivo impiegando le risorse minime indispensabili

Corso di Laurea in Informatica, Università di Padova
12/40



Introduzione

## Glossario – 4

❑ **Prodotto SW**

- **Commessa**
  - Con forma, contenuto e funzione fissate dal committente
- **Pacchetto**
  - Con forma, contenuto e funzione idonee alla replicazione
- **Componente**
  - Con forma, contenuto e funzione adatte alla composizione
- **Servizio**
  - Con forma, contenuto e funzione fissate dal problema

Corso di Laurea in Informatica, Università di Padova

13/40



Introduzione

## SWE rispetto alle altre discipline

❑ *The scientific disciplines of reference to SWE include **not only** computer science but also certain areas of discrete mathematics and operation research, statistics, psychology and economics*

❑ *SWE is **not** a branch of computer science; it is an engineering discipline relying in part on computer science, in the same way that mechanical engineering relies on physics*

Fonte: Lionel Briand, IEEE Software 49(4), 93-95

Corso di Laurea in Informatica, Università di Padova

15/40



Introduzione

## Glossario – 5

❑ **Ciclo di vita [dello sviluppo SW]**

- Gli stati che lo sviluppo del prodotto SW richiesto, assume dal concepimento al ritiro



❑ **Best practice**

- Modo di fare (*way of working*) noto, che abbia mostrato di garantire i migliori risultati in circostanze note e specifiche

Corso di Laurea in Informatica, Università di Padova

14/40



Introduzione

## SWE rispetto a se stessa

❑ **Un sistema SW è tanto più utile quanto più è usato**

- **Metrica:** integrale degli usi (o #utenti) nel tempo

❑ **Prodotti con ciclo di vita "lungo" hanno costi dominati dalla manutenzione**

- **Manutenzione correttiva** → rimozione di difetti
- **Manutenzione adattativa** → raffinamento di requisiti
- **Manutenzione evolutiva** → evoluzione del sistema

❑ **Per contenere tali costi (metrica: efficienza) serve SW manutenibile**

- Cosa significa? Come garantirlo? Ne tratteremo più avanti

Corso di Laurea in Informatica, Università di Padova

16/40



Introduzione

## Cos'è l'ingegneria del software – 1

- ❑ Nasce nel 1968 → ha compiuto 50 anni
  - Conferenza NATO 7-11/10 @ Garmisch (D)
- ❑ Raccogliere, organizzare e consolidare conoscenza (*body of knowledge*) necessaria a realizzare progetti SW con massima efficienza ed efficacia
  - Acquisizione, utilizzo e manutenzione di *best practice*
- ❑ In sostanza, l'applicazione di principi ingegneristici alla produzione del SW



Corso di Laurea in Informatica, Università di Padova

17/40



Introduzione

## Cosa facciamo in questo corso – 2

- ❑ Studiamo tutte le attività di progetto
  - Prevalentemente nel I semestre
- ❑ Proviamo a metterle in pratica
  - Nel progetto didattico, tra fine novembre e fine aprile
- ❑ Verifichiamo l'apprendimento
  - In itinere: tramite revisioni di avanzamento
  - In fine: tramite una prova scritta individuale

Corso di Laurea in Informatica, Università di Padova

19/40



Introduzione

## Cos'è l'ingegneria del software – 2

*L'approccio sistematico, disciplinato e quantificabile allo sviluppo, l'uso, la manutenzione e il ritiro del SW*  
 Fonte: Glossario IEEE

- ❑ Approccio sistematico
  - Modo di lavorare metodico e rigoroso
  - Che studia, usa ed evolve le *best practice* di dominio
- ❑ Disciplinato
  - Che segue regole fissate
- ❑ Quantificabile
  - Che permette di misurare efficienza ed efficacia

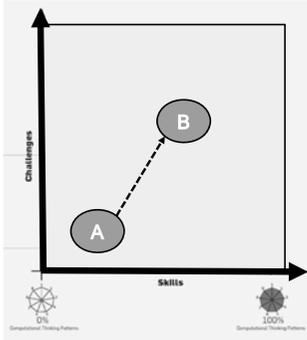
Corso di Laurea in Informatica, Università di Padova

18/40



Introduzione

## Perché lo facciamo così – 1



- ❑ *Student acquisition of computational thinking skills advances in response to challenges*
- ❑ *Pedagogical approaches can be described as instructional trajectories connecting a skill/challenge starting point (A) with a destination point (B)*

Fonte: D.C. Webb, A. Repenning, K.H. Koh, "Toward an emergent theory of broadening participation in computer science education", Proc. 43<sup>rd</sup> ACM Computer Science Education symposium, 173-178 (SIGCSE '12)

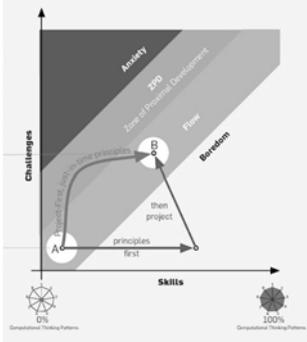
Corso di Laurea in Informatica, Università di Padova

20/40



Introduzione

## Perché lo facciamo così – 2



- ❑ *The project-first just-in-time-principles approach is in the Zone of Proximal Flow (ZPF)*
- ❑ *The Flow is the ideal condition for learning*
- ❑ *The ZPF orchestrates students' take in best practices with assistance and tool use*

Corso di Laurea in Informatica, Università di Padova

21/40



Introduzione

## Regole e vincoli di progetto – 1

- ❑ **Svolgere attività strettamente collaborative**
  - 7 persone per gruppo → condividere, ripartire, coordinare, verificare
- ❑ **Cercare soluzioni sostenibili a problemi complessi**
  - Tipologia utenti, dominio d'uso, risorse disponibili, prospettive
  - Auto-apprendimento di tecnologie e metodi di lavoro
- ❑ **Adottare un approccio ingegneristico**
  - Lavorare in modo disciplinato, sistematico, quantificabile
  - [85..105] ore di impegno individuale → costo esterno rendicontato per attività obbligatorie
  - ≈ 45 ore di esplorazione tecnologica → costo interno per attività integrative (da condividere, ripartire e contenere)

Corso di Laurea in Informatica, Università di Padova

23/40



Introduzione

## Con quale quantità di impegno

- ❑ **13 crediti → 325 ore di lavoro individuale**
- ❑ **75 ore le impegniamo in lezioni, esercitazioni e seminari**
  - 64 di esse nel I semestre
- ❑ **150 le impegniamo nel progetto didattico**
  - Fino a 105 ore di impegno in attività obbligatorie
  - Le altre 45 per auto-formazione su strumenti e metodi di lavoro rilevanti per il progetto
- ❑ **Le rimanenti 100 per studio personale in preparazione alla prova scritta individuale**
  - Studiando con *zero-latency* e non *zero-laxity*




Corso di Laurea in Informatica, Università di Padova

22/40



Introduzione

## Regole e vincoli di progetto – 2

- ❑ **Partecipa solo chi ha soddisfatto le propedeuticità**
  - Basi di Dati (superamento completo)
  - Programmazione a oggetti (almeno prova scritta)
- ❑ **Chi ha altri "arretrati", li sani prima di cimentarsi con il progetto**
- ❑ **I gruppi sono formati in sessione pubblica dopo la pubblicazione dei capitolati**
  - Gli aventi diritto si registrano in un tabellone @ Google Docs
  - Gli altri studenti si registrano in altro tabellone, specificando le propedeuticità non soddisfatte
    - Per questi verranno ulteriori regole e scadenze di ingresso



Corso di Laurea in Informatica, Università di Padova

24/40

Introduzione

## Regole e vincoli di progetto – 3

- ❑ **L'impegno necessario per raggiungere gli obiettivi di progetto deve essere superiormente limitato**
  - E deve integrarsi con tutti gli altri obblighi personali
  - Questo sconsiglia la partecipazione con "arretrati"
- ❑ **Gli obiettivi di progetto devono essere fissati in modo elastico**
  - Tra un minimo accettabile e un massimo ambizioso negoziati dinamicamente con il proponente

Corso di Laurea in Informatica, Università di Padova

25/40

Introduzione

## Gli argomenti che tratteremo

<ul style="list-style-type: none"> <li>❑ Processi, ciclo di vita e modelli di sviluppo del SW</li> <li>❑ Gestione di progetto</li> <li>❑ Amministrazione IT</li> <li>❑ Analisi dei requisiti</li> <li>❑ Progettazione</li> <li>❑ Documentazione</li> <li>❑ Qualità</li> <li>❑ Verifica e validazione</li> </ul>	<ul style="list-style-type: none"> <li>❑ UML: diagrammi dei casi d'uso</li> <li>❑ UML: diagrammi delle classi e dei <i>package</i></li> <li>❑ UML: diagrammi di sequenza e di attività</li> <li>❑ <i>Design pattern</i>: creazionali, comportamentali, architetturali</li> <li>❑ Stili architetturali</li> <li>❑ Principi SOLID</li> </ul>
---	--

Corso di Laurea in Informatica, Università di Padova

27/40

Introduzione

## Regole e vincoli di progetto – 4

Nominativo	Ore per ruolo						Ore totali
	PM	AM	AN	PT	PR	VE	
Rendicontate	8	0	6	41	24	26	105
Totale	8	4	16	41	32	40	141
Rendicontate	4	3	6	32	16	44	105
Totale	9	3	12	32	24	61	141
Rendicontate	4	8	0	36	23	34	105
Totale	12	14	11	36	31	36	140
Rendicontate	3	5	7	42	14	34	105
Totale	3	6	15	42	22	53	141
Rendicontate	3	0	6	18	30	48	105
Totale	10	2	15	18	38	57	140
Rendicontate	5	5	5	25	23	42	105
Totale	5	5	17	25	31	58	140
Rendicontate	10	6	0	38	20	31	105
Totale	10	12	19	38	28	33	140

**Dati 2016-2017**

	Re	Am	An	Pi	Pr	Ve	Tot
	7	7	14	30(+2)	24(+0)	64(+1)	146(+4)
	7	12	18(+0)	41	22(-2)	41	141(-1)
	12	13	14	18(-2)	28(+1)	61(+5)	146(+4)
	16	9	12	42(+2)	31(+1)	36(+1)	146(+4)
	15	12	19	49(+2)	19(+1)	32(+1)	146(+4)
	9	12	24	33(+2)	26(-3)	42(+5)	146(+4)
	13(+0)	13	16	24(+6)	39(+5)	41(-8)	146(+4)
Tot in ore	79(+0)	78	107(+0)	237(+12)	189(+4)	317(+5)	1071(+23)

7 persone con impegno totale complessivo individuale ≤ 150h

Attività obbligatorie

Con auto-formazione

Ore totali

Corso di Laurea in Informatica, Università di Padova

26/40

Introduzione

## Come lo facciamo – 2

- ❑ **Tre lezioni in modalità *flipped classroom***

Flipping the classroom means that students gain first exposure to new material **outside** of class (studying documents and online resources) and then use class time to do the harder work of assimilating that knowledge through problem-solving, discussion, debates

Corso di Laurea in Informatica, Università di Padova

28/40

UniPD - 2018 - Ingegneria del Software



Introduzione

## Come lo facciamo – 3

- ❑ **Strumenti di lavoro collaborativo (15 ottobre)**
  - Gestione progetto (calendari, canali, *ticket*)
  - In concerto con l'insegnamento TOS
    - Versionamento (p.es., git, SVN)
    - Configurazione e *build* (p.es, Maven, Ant, Gradle, ...)
- ❑ **Lessons learned (22 novembre)**
  - Ricercare ed elaborare "consigli dei veterani" su come organizzare la gestione collaborativa delle attività di documentazione
- ❑ **Ways of working (6 dicembre)**
  - SEMAT, sviluppo *agile* (p.es. Scrum)

Corso di Laurea in Informatica, Università di Padova

29/40



Introduzione

## Fonti e risorse – 2

- ❑ **Faremo anche riferimento a**
  - Guide to the Software Engineering Body of Knowledge (SWEBOK v3)  
IEEE Computer Society  
Software Engineering Coordinating Committee
  - <https://www.computer.org/web/swebok/>
- ❑ **Che ci aiuta a familiarizzarci con le aree di conoscenza della disciplina SWE**

Corso di Laurea in Informatica, Università di Padova

31/40



Introduzione

## Fonti e risorse – 1

- ❑ **I libri di SWE si dividono in due categorie**
  - **Teorici** : trattano la materia in modo privo di riflessi di **esperienza concreta**
    - Principi esposti, ma spesso non vissuti
  - **Esperienziali** : espongono l'esperienza dell'autore, senza **relazionarla bene alla visione generale del problema e della disciplina**
    - Eccessiva enfasi sugli aspetti accidentali
- ❑ **Noi useremo**
  - *Software Engineering*, 10th ed., 2014, di Ian Sommerville, edito da Addison Wesley (Pearson Education)

Corso di Laurea in Informatica, Università di Padova

30/40



Introduzione

## Fonti e risorse – 3

Table I.1. The 15 SWEBOK KAs	
Software Requirements	←
Software Design	←
Software Construction	←
Software Testing	←
Software Maintenance	←
Software Configuration Management	←
Software Engineering Management	←
Software Engineering Process	←
Software Engineering Models and Methods	←
Software Quality	←
Software Engineering Professional Practice	←
Software Engineering Economics	←
Computing Foundations	←
Mathematical Foundations	←
Engineering Foundations	←

Noi ci occupiamo di queste 10

Corso di Laurea in Informatica, Università di Padova

32/40



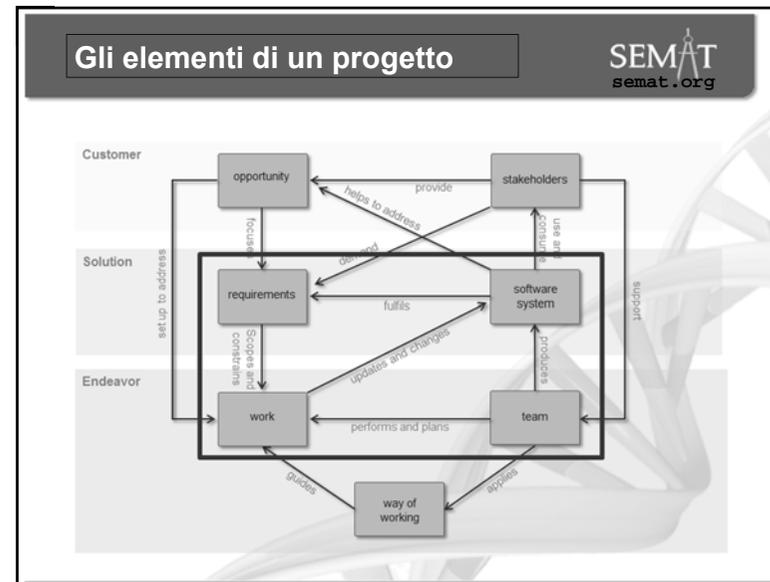
Introduzione

## Fonti e risorse – 4

- ❑ **Come altri testi di consultazione useremo**
  - **E. Gamma, R. Helm, R. Johnson, J. Vlissides**  
*Design Patterns, 2002*  
Addison-Wesley (Pearson Education Italia)
    - E le moltissime risorse internet che li approfondiscono
  - **Luciano Baresi, Luigi Lavazza, Massimiliano Pianciamore**  
*Dall'idea al codice con UML2, 2006*  
Addison-Wesley (Pearson Education Italia)

Corso di Laurea in Informatica, Università di Padova

33/40





Introduzione

## Come si studia SWE

- ❑ **Costruendo incrementalmente il proprio glossario**
  - **Basandolo inizialmente sulla teoria**
    - Individuazione dei termini, definizione dei significati
  - **Consolidandolo con la pratica**
    - Applicazione dei significati, confronto critico con l'esperienza
  - **Discutendolo con i colleghi**
    - Unendo conoscenze parziali, correggendosi reciprocamente
- ❑ **Integrando le diapositive con studio personale**
  - Ricercando altre fonti e risorse autorevoli

Corso di Laurea in Informatica, Università di Padova

34/40



Introduzione

## Glossario – 6

- ❑ **Stakeholder (portatore di interesse)**
  - **L'insieme di coloro che a vario titolo hanno influenza sul prodotto, sul progetto, sui processi**
    - La comunità degli utenti (che usa il prodotto)
    - Il committente (che compra il prodotto)
    - Il fornitore (che sostiene i costi di realizzazione)
    - Eventuali regolatori (che verificano l'attuazione di processi)
- ❑ **Way of working → i processi**
  - **La maniera di rendere sistematiche, disciplinate e quantificabili le attività di progetto**

Corso di Laurea in Informatica, Università di Padova

36/40



Introduzione

## Le 4 P del SWE

Di questi parleremo ampiamente nelle prossime lezioni

**People**  
(Gli stakeholder e il team)

**Product**  
(SW e documentazione) ✓

**Project** ✓  
(L'insieme delle attività di produzione)

**Process**  
(Way of working)

Corso di Laurea in Informatica, Università di Padova

37/40



Introduzione

## Figure professionali – 2

- **Il *software engineer***
  - **Realizza parte di un sistema complesso con la consapevolezza che potrà essere usato, completato e modificato da altri**
  - **Deve comprendere il quadro generale in cui si colloca il sistema cui contribuisce**
    - La dimensione “sistema” include ma non si limita al SW
  - **Deve operare compromessi intelligenti e lungimiranti tra visioni e spinte contrapposte**
    - Costi – qualità
    - Risorse (tempo, mezzi e competenze) – disponibilità
    - Usabilità

Corso di Laurea in Informatica, Università di Padova

39/40



Introduzione

## Figure professionali – 1

- ***Software engineer* ≠ programmatore**
- **Il programmatore**
  - **Figura professionale dominante nei primi decenni dell'informatica**
  - **Scrive programmi da solo, sotto la propria responsabilità tecnica**
  - **Svolge un'attività creativa fortemente personalizzata**



Corso di Laurea in Informatica, Università di Padova

38/40



Introduzione

## Principi etici della professione



Association for Computing Machinery

**Software Engineering Code of Ethics and Professional Practice (Short Version)**

Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:

1. PUBLIC - Software engineers shall act consistently with the public interest.
2. CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
3. PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
4. JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.
5. MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6. PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7. COLLEAGUES - Software engineers shall be fair to and supportive of their colleagues.
8. SELF - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Corso di Laurea in Informatica, Università di Padova

40/40