



Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	Integrazione e Test di Sistemi Software	
Corso di studio	Informatica e Tecnologie per la Produzione del Software	
Anno Accademico	2023/24	
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS)	6 CFU	
Settore Scientifico Disciplinare	ING-INF/05	
Lingua di erogazione	Italiano	
Anno di corso	Terzo	
Periodo di erogazione	1^ semestre, le date esatte sono riportate nel manifesto/regolamento	
Obbligo di frequenza	La frequenza è fortemente raccomandata	
Sito web del corso di studio	https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/informatica-tps-270/laurea-triennale-in-informatica-e-tecnologie-per-la-produzione-del-software-d.m.-270	

Docente/i	
Nome e cognome	Azzurra Ragone
Indirizzo mail	azzurra.ragone@uniba.it
Telefono	+39 080-5443289 (int. 3289)
Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n.616, 6^ piano.
Sede virtuale	Piattaforma ADA - https://elearning.di.uniba.it/
Sito web del docente	https://www.uniba.it/it/docenti/ragone-azzurra
Ricevimento (giorni, orari e modalità, es. su appuntamento)	Giovedì 16:00 – 17:30 (su appuntamento da concordare per email con il docente)

Syllabus



Obiettivi formativi	L'insegnamento si propone di introdurre le metodologie, strategie, tecniche e strumenti di integrazione e testing che concorrono alla Verifica e Validazione del software. Obiettivo dell'insegnamento è favorire l'acquisizione di competenze utili sia allo sviluppo di software di qualità che per la sua valutazione.
Prerequisiti	Le seguenti conoscenze preliminari facilitano ed accelerano la comprensione degli argomenti dell'insegnamento: - da Programmazione II: fondamenti della programmazione ad oggetti (Java), gestione delle eccezioni in Java, lambda expression e Stream, familiarità di utilizzo di un ambiente di sviluppo (un IDE, tipo Eclipse, IntelliJ, ecc.).
Contenuti di insegnamento (Programma)	<ul style="list-style-type: none">- Introduzione al Software Testing: Validazione e Verifica – Terminologia - Come selezionare le tipologie di test più adeguate - Test automation – La piramide dei test- Testing Black Box. Testing basato sui requisiti e sugli scenari dei casi d'uso - Test di tipo funzionale – identificare test case per i valori soglia (boundary test)- Testing White Box Code coverage - criteri di code coverage - Strumenti per la misura automatica della copertura del codice - Test di tipo strutturale- Design Contracts Definire le pre-condizioni e post-condizioni nei contratti, differenza tra contratti e validazione del software- Property-based testing Cosa sono e differenza con i test di tipo strutturale e funzionale- Mocking framework Dummies, fakes, stubs, spies e mocks - Test double- Test-driven development (TDD) Cosa si intende per TDD - Quando usare l'approccio TDD e quando non usarlo- Qualità del codice di test Come scrivere codice di qualità e che sia manutenibile – Test smell- Documentazione di test Documento di strategia dei test – Test Plan – Test Design Specification Document- Integration and System Test Testare diversi componenti del software – Testing di SQL e Database – Test di Sistemi- Analisi costi/benefici- JUnit: Introduzione a JUnit - Implementazione di test di unità con JUnit su programmi Java – Assunzioni e asserzioni – Testing delle eccezioni – Test dinamici e parametrici – Testing Data Driven con JUnit
Testi di riferimento	<p><u>Testo di riferimento:</u> Effective Software Testing (A developer's guide) Mauricio Aniche. Manning. ISBN 9781633439931, 2022 (NB: Vengono trattati tutti i capitoli del libro)</p> <p><u>Testi consigliati:</u> Pezzè, Mauro, Young, Michal. Software testing and analysis: process, principles and techniques. Hoboken, N.J.: Wiley, 2008. G. J. Myers, T. Badgett, C. Sandler. The Art of Software Testing. Wiley (Cap.1, 2, 4, 20, 21, 22, 24)</p> <p>Materiale messo a disposizione dal docente</p> <p>Gli studenti che lo desiderano possono ottenere i testi in prestito dalla Biblioteca. Può convenire verificarne la disponibilità mediante il Sistema Bibliotecario di Ateneo https://opac.uniba.it/easyweb/w8018/index.php? e contattare la biblioteca per concordare il prestito.</p>
Note ai testi di riferimento	Sono disponibili su ADA:



	<ul style="list-style-type: none"> - Slide usate a lezione (caricate dopo ogni lezione) - Codici degli esercizi svolti durante i laboratori <p>Il docente indica sempre alla fine delle slide di ogni lezione il testo e il capitolo del libro a cui la lezione fa riferimento. Inoltre, sempre alla fine delle slide vengono indicati i riferimenti ad eventuali altri materiali bibliografici (white paper, articoli scientifici, guide, ecc.), repository di codice ed eventuali approfondimenti.</p> <p>In relazione al Caso di Studio, lo stesso va concordato con il docente e deve essere aderente agli argomenti del corso.</p>		
Organizzazione della didattica			
Ore			
Totali	Didattica frontale	Pratica (laboratorio, progetto, esercitazione, altro)	Studio individuale
150 ore	32 ore	30 ore	88 ore
CFU/ETCS			
6 CFU	4 CFU	2 CFU	

Metodi didattici	
	<ul style="list-style-type: none"> ▪ Lezioni frontali condotte con l'ausilio di slide proiettate in aula e rese disponibili tramite la piattaforma di e-learning; ▪ Svolgimento in aula di due tipologie di esercitazione: (a) esercizi svolti interamente dal docente con indicazione delle soluzioni; (b) esercitazioni guidate in cui gli studenti risolvono da soli, ma supervisionati dal docente, problemi relativi al testing di sistemi software. <p>Entrambe le tipologie di esercitazione sono svolte con l'obiettivo di acquisire dimestichezza con le metodologie di software testing da applicare successivamente al caso di studio da svolgere preferibilmente in gruppo.</p>

Risultati di apprendimento previsti	
Conoscenza e capacità di comprensione	<ul style="list-style-type: none"> ▪ Comprensione del concetto di qualità del software e dei concetti di verifica e validazione dello stesso ▪ Conoscenza delle diverse tecniche di integrazione e testing dei sistemi software (ad es. white-box e black-box), metodologie di sviluppo (ad es. sviluppo test driven) e degli strumenti a supporto degli sviluppatori per l'integrazione e testing di tali sistemi



Conoscenza e capacità di comprensione applicate	<ul style="list-style-type: none">▪ Saper applicare le best practice del software testing per garantire lo sviluppo di software di qualità (robusto, affidabile, ecc.).▪ Saper applicare i concetti, le tecniche e le metodologie di integrazione e testing di sistemi software, nonché utilizzare gli strumenti di supporto agli sviluppatori.
Competenze trasversali	<p>Autonomia di giudizio</p> <p>1) Acquisire autonomia di giudizio sulle scelte da effettuarsi relativamente allo sviluppo di test case suite e alle tecniche di integrazione e testing da adottare.</p> <p>Abilità comunicative</p> <p>2) Capacità di comunicare sia in maniera orale che tramite documenti scritti (ad es. test case suite) la soluzione proposta; saper modulare la comunicazione rispetto ai diversi stakeholder (tecnici e non) di un progetto software usando la giusta terminologia.</p> <p>Capacità di apprendere in modo autonomo</p> <p>3) Sviluppare capacità di intraprendere in autonomia ulteriori approfondimenti su argomenti attinenti all'integrazione e testing di sistemi software.</p>

Valutazione	
Modalità di verifica dell'apprendimento	<p>La verifica dei risultati formativi raggiunti avviene durante l'esame finale, che prevede un colloquio orale in cui:</p> <ul style="list-style-type: none">▪ Si presenta e si discute in maniera critica il caso di studio sviluppato in gruppo;▪ Si verificano le competenze acquisite durante il corso;▪ Si accertano le capacità espositive dello studente e la capacità di motivare le scelte progettuali. <p>Il caso di studio va concordato con il docente e inizia ad essere sviluppato sotto la guida del docente durante le ore di laboratorio.</p>
Criteri di valutazione	<ul style="list-style-type: none">• Conoscenza e capacità di comprensione: Si valuta attraverso il caso di studio, e la successiva discussione nella prova orale, la conoscenza e la capacità di comprensione da parte dello studente dei concetti, delle tecniche, degli strumenti e delle metodologie di integrazione e testing di sistemi software affrontati a lezione• Conoscenza e capacità di comprensione applicate:



	<p>Si valuta attraverso il caso di studio, e la successiva discussione nella prova orale, la capacità di saper applicare le best practice per garantire lo sviluppo di software di qualità, oltre a saper applicare i concetti, le tecniche e le metodologie di integrazione e testing di sistemi software, nonché utilizzare gli strumenti di supporto</p> <ul style="list-style-type: none">• Autonomia di giudizio: Si valuta attraverso il caso di studio, e la successiva discussione nella prova orale, la capacità di effettuare autonomamente scelte relativamente allo sviluppo di test case suite e alle tecniche di integrazione e testing da adottare.• Abilità comunicative: Si valuta attraverso il caso di studio e la successiva discussione nella prova orale la capacità di comunicare la soluzione proposta usando la giusta terminologia• Capacità di apprendere: Si valuta attraverso il caso di studio e la successiva discussione nella prova orale la capacità di applicare i concetti appresi durante il corso ad un caso di studio nuovo non analizzato a lezione
Criteri di misurazione dell'apprendimento e di attribuzione del voto finale	Il voto finale è dato dalla media aritmetica dei voti conseguiti per l'elaborazione del caso di studio e prova orale arrotondato per eccesso. Il caso di studio pesa per il 40% del voto complessivo e la prova orale per l'altro 60%.
Altro	<p>Si suggerisce agli studenti di affidarsi esclusivamente alle informazioni/comunicazioni fornite sui siti ufficiali del Dipartimento di Informatica, ovvero sui gruppi social solo se costituiti e amministrati esclusivamente dai docenti dei relativi insegnamenti:</p> <ul style="list-style-type: none">• https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea• https://www.uniba.it/it/ricerca/dipartimenti/informatica• https://elearning.di.uniba.it/ <p>I programmi degli insegnamenti sono disponibili qui:</p> <ul style="list-style-type: none">• https://programmi.di.uniba.it/ <p>Le informazioni che tutti gli studenti dovrebbero conoscere sono scritte nei Regolamenti didattici e manifesti degli studi disponibili nel sito:</p> <ul style="list-style-type: none">• https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea <p>Si suggerisce agli studenti di diffidare delle informazioni e dei materiali circolanti su siti o gruppi social non ufficiali, poiché spesso sono risultati non affidabili, non corretti o incompleti. Per ogni dubbio, chiedere un incontro al docente secondo le modalità previste per il ricevimento.</p> <hr/> <p>Link al corso sulla piattaforma e-learning del dipartimento ADA: https://elearning.di.uniba.it/course/view.php?id=1086</p>