Logo Image not found or type unknown

PROGRAMMA DEL CORSO DI INGEGNERIA DEL SOFTWARE

SETTORE SCIENTIFICO

ING-INF/05

CFU

9

MODALITÀ DI ESAME ED EVENTUALI VERIFICHE DI PROFITTO IN ITINERE

Modalità di iscrizione e di gestione dei rapporti con studenti

L’iscrizione ed i rapporti con gli studenti sono gestiti gli mediante la piattaforma informatica che permette l’iscrizione ai corsi, la fruizione delle lezioni, la partecipazione a forum e tutoraggi, il download del materiale didattico e la comunicazione con il docente. Un tutor assisterà gli studenti nello svolgimento di queste attività.

Attività di didattica erogativa

54 Videolezioni + 54 test di autovalutazione
Impegno totale stimato: 54 ore

Attività di didattica interattiva

Redazione di un elaborato su traccia del docente Partecipazione a una web conference Svolgimento delle prove in itinere con feedback Svolgimento della simulazione del test finale
Impegno totale stimato: 9 ore

Attività di autoapprendimento

162 ore per lo studio individuale

RISULTATI DI APPRENDIMENTO ATTESI

Conoscenza e capacità di comprensione.

Il corso intende fornire le conoscenze utili per la comprensione delle caratteristiche essenziali del software, dei processi software e dei principali cicli di produzione e di vita del software, sia tradizionali che agili. Conoscenza e comprensione di dettaglio: del Processo di Elicitazione e Analisi dei Requisiti di Sistemi Software con modelli, metodi e tecniche ad oggetti; del Processo di Progettazione Software via patterns. Conoscenze preliminari di manutenzione e testing del

software.

Capacità di applicare conoscenza e comprensione.

Il corso trasferisce la capacità di definire i requisiti e progettare sistemi software di piccole dimensioni; capacità di lavorare in team nelle diverse fasi di un ciclo di produzione di software. Capacità di usare framework, tools e piattaforme tecnologiche per la produzione di software.

Autonomia di giudizio.

Attraverso le competenze acquisite, lo studente avrà autonome capacità di giudizio su qualità di requisiti (completezza, correttezza, verificabilità, coerenza, etc..) e progetto; autonoma capacità di giudicare e comparare tecnologie per la produzione di software.

Abilità comunicative.

Lo studente svilupperà la capacità di interagire e comunicare in team di produzione software sia in cicli agili che tradizionali. Capacità di comunicare con tutti gli stakeholders che concorrono a definire fabbisogno e requisiti software. Capacità di apprendimento. La capacità di apprendimento sarà stimolata attraverso la somministrazione di esercitazioni operative, caricate in piattaforma nella sezione elaborati, finalizzata anche a verificare l'effettiva comprensione degli argomenti trattati. Lo studente acquisirà, inoltre, la capacità di seguire l'evoluzione di metodi, tecniche e tecnologie della Ingegneria del Software, di seguirne i trend di mercato ed applicativi attraverso report, standard e letteratura tecnica di settore. Capacità di autoaggiornarsi attraverso documentazione tecnica, selezione ed uso di courseware, addestramento su nuove piattaforme tecnologiche.

MODALITÀ DI RACCORDO CON ALTRI INSEGNAMENTI

Il corso si raccorda in particolare al corso di Ingegneria dei dati. Il raccordo avverrà tramite la preliminare condivisione del programma tra i docenti finalizzata ad evitare duplicazioni/sovrapposizioni del programma ed assicurare la completezza degli argomenti trattati

MODALITÀ DI ESAME ED EVENTUALI VERIFICHE DI PROFITTO IN ITINERE

/**/
Lo studente per superare l'esame può scegliere di effettuare l'esame orale presso la sede dell'Ateneo o la prova scritta in tutte le sedi di Italia, ivi compreso Roma.

Il test finale si compone di 31 domande a risposta multipla con 4 possibili risposte.

Le domande di esame siano esse orali o scritte, coerentemente con i risultati di apprendimento attesi, sono finalizzate a misurare la preparazione acquisita in relazione a

Conoscenza e capacità di comprensione attraverso domande sul programma del corso Capacità di applicare conoscenza e comprensione attraverso domande specifiche che consentano la valutazione rispetto a casi concreti Autonomia di giudizio attraverso domande che presuppongano la valutazione autonoma in ordine alla scelte da compiere.
Gli esercizi e gli elaborati di Didattica erogativa consentono invece di verificare i risultati di apprendimento raggiunti rispetto alle abilità comunicative e alla capacità di apprendimento.

OBIETTIVI FORMATIVI PER IL RAGGIUNGIMENTO DEI RISULTATI DI APPRENDIMENTO PREVISTI

Il corso propone una serie di argomenti per conoscere e definire i fondamentali aspetti architetturali dei moderni sistemi software. L'obiettivo del corso è quello di fornire una comprensione approfondita dei concetti del paradigma object-oriented e di fornire gli elementi per la progettazione di applicazioni software con metodologie orientate agli oggetti.

PROGRAMMA DIDATTICO

1 - INTRODUZIONE ALL'INGEGNERIA DEL SOFTWARE 2 - CONCETTI GENERALI DELL'INGEGNERIA DEL SOFTWARE 3 - INTRODUZIONE AL CICLO DI VITA DEL SOFTWARE 4 - I MODELLI DEL CICLO DI VITA DEL SOFTWARE 5 - ORGANIZZAZIONE E COMUNICAZIONE NEI PROGETTI SOFTWARE 6 - UNIFIED MODELING LANGUAGE: INTRODUZIONE 7 - DIAGRAMMI DI CASI D'USO 8 - DESCRIZIONE CASI D'USO E ESERCITAZIONE 9 - DIAGRAMMA DELLE CLASSI 10 - MODELLARE LE RELAZIONI FRA CLASSI 11 - DIAGRAMMI DI SEQUENZA 12 - DIAGRAMMA DI SEQUENZA: ESERCITAZIONE 13 - DIAGRAMMI DI MACCHINA A STATI 14 - DIAGRAMMI DI MACCHINA A STATI: ESERCITAZIONE 15 - DIAGRAMMI DI ATTIVITA' 16 - DIAGRAMMI DELLE ATTIVITA': CONCETTI AVANZATI E ESERCITAZIONE 17 - USARE I DIAGRAMMI UML 18 - DIAGRAMMI UML: ESERCITAZIONE 19 - APPROCCIO DI SVILUPPO ORIENTATO AGLI OGGETTI 20 - PRINCIPI E CONCETTI OBJECT-ORIENTED 21 - APPROCCIO ORIENTATO AGLI OGGETTI NEL PROCESSO DI PRODUZIONE SOFTWARE 22 - JAVA: INTRODUZIONE AL LINGUAGGIO 23 - JAVA: COMPONENTI FONDAMENTALI DI UN PROGRAMMA 24 - JAVA: VARIABILI, COSTRUTTORI E PACKAGE 25 - IDENTIFICATORI E TIPI DI DATI PRIMITIVI E COMPLESSI 26 - JAVA: OPERATORI E ARRAY 27 - JAVA: COSTRUTTI DI CICLI E CONDIZIONI 28 - OBJECT ORIENTED IN JAVA: INCAPSULAMENTO E VISIBILITA' 29 - OBJECT ORIENTED IN JAVA: VISIBILITA' E EREDITARIETA' 30 - OBJECT ORIENTED IN JAVA: VISIBILITÀ E EREDITARIETÀ 31 - ESERCITAZIONE JAVA - INTRODUZIONE 32 - ESERCITAZIONE JAVA - VARIABILI, COSTRUTTORI, OPERATORI E ARRAY 33 - ESERCITAZIONE JAVA - ARRAY, COSTRUTTI DI CICLI E CONDIZIONE 34 - ESERCITAZIONE JAVA :INCAPSULAMENTO 35 - ESERCITAZIONE JAVA : EREDITARIETÀ 36 - ESERCITAZIONE JAVA : POLIMORFISMO 37 - DESIGN PATTERN: PANORAMICA E ITERATOR 38 - ESERCITAZIONI - DESIGN PATTERN: SINGLETON 39 - ESERCITAZIONI - DESIGN PATTERN: COMPOSITE 40 - ESERCITAZIONI - DESIGN PATTERN: STATE 41 - PROCESSI SOFTWARE E MODELLI DI SVILUPPO 42 - INGEGNERIA DEI REQUISITI 43 - INGEGNERIA DELLA PROGETTAZIONE 44 - TEST DEL SOFTWARE 45 - STRUMENTI PER IL DEVOPS 46 - METODOLOGIA DEVOPS 47 - CONTROLLO DI VERSIONE 48 - GIT 49 - ESERCITAZIONE GIT 50 - AUTOMAZIONE 51 - LINGUAGGI DICHIARATIVI PER L'AUTOMAZIONE 52 - CONTAINER 53 - DOCKER 54 - ESERCITAZIONE DOCKER