

Testi del Syllabus

Resp. Did.	MONTELLA RAFFAELE	Matricola: 001713
Docente	MONTELLA RAFFAELE, 6 CFU	
Anno offerta:	2022/2023	
Insegnamento:	TW6 - TECNOLOGIE WEB CFU 6	
Corso di studio:	0124 - INFORMATICA	
Anno regolamento:	2020	
CFU:	6	
Settore:	INF/01	
Tipo Attività:	D - A scelta dello studente	
Anno corso:	3	
Periodo:	Primo Semestre	



Testi in italiano

Lingua insegnamento	Italiano. Se presente almeno uno studente Erasmus, il corso sarà tenuto in inglese.
Contenuti	Le tecnologie web Introduzione al cloud computing Introduzione al linguaggio Python I web service . I client per il web Linguaggi di descrizione per le pagine web Linguaggi di scripting Introduzione ai database NoSQL e a MongoDB Trattamento dei dati geografici Internet of things
Testi di riferimento	Programmazione Web Lato Server - ultima versione aggiornata - V. Della Mea, L. Di Gaspero, I. Scagnetto. Apogeo Learning Python - M. Lutz. O'Reilly Flask Web Development: Developing Web Applications with Python - M. Grimberg. O'Reilly Cloud computing per applicazioni web - J. Barr. Apogeo Il manuale di Arduino: guida completa - P. Aliverti, LSWR Le slide (formato .pdf) di tutte le lezioni sono disponibili sulla piattaforma di e-learning del Dipartimento di Scienze e Tecnologie, insieme con quiz on-line di autovalutazione, esercizi, progetti di approfondimento, note per il laboratorio, una nota di introduzione operativa agli strumenti software trattati. In particolare, per ogni edizione del corso, le slide sono prodotte in ambiente Google Drive con condivisione con gli studenti che possono arricchire il materiale con contenuti specifici. Tutto il materiale prodotto in questa forma, nel rispetto dei diritti delle singole fonti, è rilasciato con licenza Creative Commons.
Obiettivi formativi	Il corso è un'introduzione alla programmazione di applicazioni in ambiente internet con riferimenti anche alla fruizione attraverso dispositivi mobili. Il corso contiene un'introduzione alla programmazione di applicazioni web mediante l'uso della piattaforma LAMP (Linux Apache MongoDB Python). Una parte del corso è dedicata all'introduzione al Cloud Computing con

particolare riferimento ai servizi offerti da Amazon Web Services per le tecnologie web e all'Internet of Things con microcontrollori di tipo Arduino o similari. Nel corso delle lezioni sono fatti continui riferimenti alle tecnologie per lo sviluppo di applicazioni mobile ibride multiplatforma, alla gestione e rappresentazione dei dati spaziali sul web e ai sistemi pervasivi distribuiti/indossabili.

Il corso contiene una introduzione al linguaggio Python, utilizzato per lo sviluppo di progetti software e hardware nelle attività (singole o di gruppo) in Laboratorio che sono parte integrante del corso. Il corso ha una impostazione tipicamente pratico/applicativa finalizzata alla risoluzione di problemi concreti e allo sviluppo di software effettivamente utilizzabile.

Conoscenza e capacità di comprensione: Lo studente deve dimostrare di conoscere e saper comprendere i fondamenti delle tecnologie web, del cloud computing e dell'Internet of Things, con particolare riguardo alle applicazioni web complesse e distribuite, le metodologie di sviluppo e di analisi di applicazioni basate su servizi web, il linguaggio di programmazione Python e il suo contesto applicativo, le metodologie e le tecniche dello sviluppo software in tale contesto.

Capacità applicative: Lo studente deve dimostrare di saper utilizzare la propria conoscenza acquisita per risolvere problemi computazionali come applicazioni web o applicazioni mobile/embedded ibride basate su tecnologie web in contesti concreti, per sviluppare soluzioni progettuali e analizzarle dal punto di vista dell'architettura e dell'usabilità, per usare in modo consapevole uno o più linguaggi di programmazione al fine di implementare un'applicazione web complessa. Tali capacità si estrinsecano anche in un uso approfondito e consapevole degli strumenti offerti dal cloud computing e dall'Internet of Things e differenti framework applicativi.

Autonomia di giudizio: Lo studente deve essere in grado di sapere valutare in maniera autonoma i risultati ottenuti dalla applicazione sviluppata sia in termini progettuale che dell'implementazione software valutandone l'efficacia e l'efficienza.

Abilità comunicative: Lo studente deve essere in grado di redigere una relazione di presentazione di un progetto e di documentare la sua implementazione, anche lavorando in gruppo, servendosi di strumenti avanzati di scrittura/documentazione con particolare riferimento all'uso degli strumenti di produttività per sviluppatori come sistemi per la redazione condivisa di documenti (Google Drive), di collaborazione remota (Slack), di gestione della produzione (Trello), di gestione dei repository online (GitHub), di infrastrutture cloud come servizio (Amazon AWS) usando correttamente la terminologia delle tecnologie web anche in lingua inglese.

Capacità di apprendimento: Lo studente deve essere in grado di aggiornarsi e approfondire in modo autonomo argomenti e applicazioni specifiche di tecnologie web, anche accedendo a banche dati, repository on-line di documenti, software e librerie applicative e altre modalità messe a disposizione dalla rete.

Prerequisiti

E' necessaria la conoscenza degli argomenti svolti nei corsi di Programmazione I, Programmazione II: il corso è CONSIGLIATO al terzo anno di corso.

Metodi didattici

Lezioni frontali
Seminari
Imitazione di casi reali
Progetti didattici

Altre informazioni

Microsoft Team:
Code: 28ktcjh
Link: <https://teams.microsoft.com/l/team/19%3a9b9c6ce8a31546b98f4c068af680d394%40thread.tacv2/conversations?groupId=ce90e83d-649f-46ab-9771-3eb3e0b7e295&tenantId=017e16ae-f415-4f8d-9af0-a21b57cd448e>

Modalità di verifica dell'apprendimento

L'obiettivo della procedura di verifica consiste nel quantificare il livello di raggiungimento degli obiettivi formativi precedentemente indicati. La procedura di verifica è indicata precisamente nella piattaforma di e-learning del Dipartimento di Scienze e Tecnologie. In sintesi: a circa 1/3 del corso gli studenti sottomettono una proposta di progetto rafforzata da una presentazione composta da non più di 3 slide e della durata non superiore ai 3 minuti. La proposta può essere individuale o di un gruppo composto da non più di 3 studenti. La proposta viene emendata dal docente, rigettata o assegnata come progetto di esame. A circa 2/3 del corso è organizzata una presentazione dei progetti in cui ciascuno studente o gruppo di studenti illustra lo stato dell'arte del progetto e discute con docenti e altri studenti di potenzialità e debolezze dello stesso. A fine corso è organizzata una presentazione finale che non ha valore di esame ma momento comunicativo del lavoro svolto. Il progetto deve essere rilasciato in open source preferibilmente con licenza Apache 2.0 su di un repository pubblico come, ad esempio, GitHub almeno 2 giorni prima della data di esame e deve essere funzionante online almeno fino alla conclusione dell'esame. La procedura di verifica consiste in un esame orale individuale (60% del voto), la presentazione della parte di progetto di propria competenza (20% del voto), dimostrazione del progetto (20% del voto). Sono previste 2 Prove Intercorso (40% del voto dell'esame orale) + 2 HomeWork (20% del voto dell'esame orale) individuali.

Programma esteso

Le tecnologie web - Il paradigma client-server - Iper testo, multimedia, ipermedia - Il web come documento ipermediale - Il protocollo HTTP, richieste, risposte, intestazioni, cookie e mantenimento delle sessioni - Server web.

Introduzione al cloud computing - infrastruttura come servizio - Virtualizzazione - immagini ed istanze di macchine virtuali - Gestire macchine virtuali in Amazon Web Service - Preparare una macchina virtuale per la produzione di servizi web - Accedere alle macchine virtuali AWS - Elementi di Linux per la gestione delle macchine virtuali e dei software per le tecnologie web.

Introduzione al linguaggio Python - costrutti, variabili, tipi di dati - Python argomenti avanzati: classi, oggetti e namespace - Python come linguaggio per le tecnologie web: introduzione a Flask - Generare pagine web con Flask.

I web service - SOAP e REST - JSON: Javascript Object notation - GET, POST e gli altri metodi HTTP in Flask.

I client per il web - il browser - altri client - le applicazioni mobile

Programmazione server side e client side

Linguaggi di descrizione per le pagine web - Extensible Markup Language - Il markup - XML come metalinguaggio - Caratteristiche sintattiche - Documenti XML ben formati - XHTML e HTML 5

Caratteristiche generali di una pagina web - Elementi ed attributi - Struttura base del documento - Testo, iper testo, immagini, liste ed elenchi, form, tabelle, collegamenti e scripting lato client - Il cascading style sheets (CSS): introduzione.

Generazione dinamica delle pagine - Gestione degli header http - File attaching - Templating in Python/Flask.

Introduzione ai database NoSQL e a MongoDB - Python & MongoDB

Trattamento dei dati geografici: Leaflet & Geojson

Introduzione ai websocket - Metodi avanzati e consumo dei websocket in JavaScript - Servire websocket in Python

Internet of things, introduzione ad Arduino e alle tecnologie web

Esercizio applicativo: "The Imitation Game" - Sviluppo di un'applicazione imitante: "Facebook", "Uber", "Open table" o similari.

Obiettivi per lo sviluppo sostenibile

Codice

Descrizione



Testi in inglese

Italian. If an Erasmus student is attending the class, the course will be taught in English.

Web Technologies
Cloud Computing
The Python Language
The Web Services
Web Clients
Web page description languages
Client side scripting
NoSQL database
Geographical Data
Internet of things

Web Technologies – U.K. Ray. Oxford Press
Learning Python – M. Lutz. O'Reilly
Flask Web Development: Developing Web Applications with Python – M. Grimberg. O'Reilly
Amazon Web Services in Action – A. Witting, M. Witting. Manning.
Programming Arduino – S. Monk, McGrawHill

The slides (.pdf format) of all lessons are available on the e-learning platform of the Department of Science and Technology, along with an on-line self-assessment quiz, exercises, deepening projects, lab notes, an introduction note Operational to the software tools handled. In particular, for each edition of the course, the slides are produced in a Google Drive-sharing environment with students who can enrich content with specific content. All material produced in this form, while respecting the rights of individual sources, is licensed under Creative Commons.

The course is an introduction to programming applications in the internet environment with references to the use of mobile devices. The course contains an introduction to programming web applications using the LAMP platform (Linux Apache MongoDB Python). Part of the course is devoted to the introduction to Cloud Computing with particular reference to the services offered by Amazon Web Services for web technologies and the Internet of Things with Arduino or similar microcontrollers. During the lessons, continuous references are made to the technologies for the development of mobile multi-platform hybrid applications, the management and representation of spatial data on the web and distributed / wearable pervasive systems.

The course contains an introduction to the Python language, used for the development of software projects and hardware in the activities (single or group) in the Laboratory that are an integral part of the course. The course has a typically practical / application setting designed to solve concrete problems and develop software that is actually usable.

Understanding skills: The student must demonstrate knowledge and understanding of the fundamentals of web technologies, cloud computing, and the Internet of things, with particular regard to complex and distributed web applications, web-based application development and analysis methodologies, Python programming language and its application context, methodologies and software development techniques in this context.

Applicative skills: The student must demonstrate his / her knowledge acquired to solve computational problems such as web applications or mobile / embedded hybrid applications based on web technologies in concrete contexts, to develop design solutions and analyze them from the point of view of architecture and usability, to consciously use one or more programming languages to implement a complex web application. These capabilities also extricate in a thorough and aware use of the tools offered by cloud computing and the Internet of things and different application frameworks.

Judgment autonomy: The student must be able to know how to

independently evaluate the results obtained from the application developed in both design and software implementation, evaluating its effectiveness and efficiency.

Communication skills: The student must be able to draw up a project submission report and document its implementation, including working in a group, using advanced writing / documentation tools, with particular reference to the use of productivity tools for developers as systems for Shared Publishing (Google Drive), Remote Collaboration (Slack), Production Management (Trello), Online Repository Management (GitHub), Cloud Infrastructure as a Service (Amazon AWS) Using the Web Technology Terminology Correctly Also in English.

Learning skills: The student must be able to update and deepen autonomously specific web technology topics and applications, including accessing databases, online document repositories, software and application libraries, and other ways available from the network.

Knowledge of the topics in Programming I, Programming II is needed: the course is SUGGESTED to be attended at the 3rd year.

Lectures
Seminaries
Imitation games
Projects

Microsoft Team:
Code: 28kctjh

Link: <https://teams.microsoft.com/l/team/19%3a9b9c6ce8a31546b98f4c068af680d394%40thread.tacv2/conversations?groupId=ce90e83d-649f-46ab-9771-3eb3e0b7e295&tenantId=017e16ae-f415-4f8d-9af0-a21b57cd448e>

The purpose of the verification procedure is to quantify the level of achievement of the training objectives previously indicated.

The verification procedure is indicated precisely in the e-learning platform of the Department of Science and Technology. In summary: around 1/3 of the course, students submit a draft proposal reinforced by a presentation consisting of no more than 3 slides and a duration not exceeding 3 minutes. The proposal may be individual or group consisting of no more than 3 students. The proposal is amended by the teacher, rejected or assigned as an exam project. About 2/3 of the course is organized with a presentation of the projects in which each student or group of students illustrates the state of the art of the project and discusses with teachers and other students about its potentialities and weaknesses.

At the end of the course is organized a final presentation that has no value of examination but a moment of communication of the work done. The project must be released in open source rather than Apache 2.0 on a public repository such as, for example, GitHub at least 2 days before the exam date and must be online at the end of the exam. The verification procedure consists of an individual oral exam (60% of the vote), the presentation of the project part of its competence (20% of the vote), project demonstration (20% of the vote). There are 2 Test Examinations (40% of the oral exam) + 2 HomeWork (20% of the oral exam) individual.

Web technologies - Client-server paradigm - Hypertext, multimedia, hypermedia - Web as a hypermedia document - HTTP protocol, requests, responses, headers, cookies and session maintenance - Web server.

Introduction to Cloud Computing - Infrastructure as a Service - Virtualization - Virtual Machines Image and Instances - Managing Virtual Machines in Amazon Web Service - Preparing Virtual Machines for Web Services Production - Accessing AWS Virtual Machines - Linux Management Elements Virtual machines and software for web technologies.

Introduction to Python Language - Buildings, Variables, Data Types - Python advanced topics: classes, objects and namespaces - Python as a web technology language: introduction to Flask - Generating web pages

with Flask.
Web Services - SOAP and REST - JSON: Javascript Object Notation - GET, POST, and other HTTP methods in Flask.
Web clients - browsers - other clients - mobile applications
Server side and client side programming
Extensible Markup Language - Markup - XML as a Template - Syntax
Features - Well-Formatted XML Documents - XHTML and HTML 5
General Features of a Web Page - Elements and Attributes - Basic
Document Structure - Text, hypertext, images, lists and lists, forms, tables, links, and client side scripting - Cascading style sheets (CSS): Introduction.
Dynamic page generating - Managing http headers - File attachching - Templating in Python / Flask.
Introduction to NoSQL and Mongodb databases - Python & MongoDB
Geographic data processing: LeatLet & GeoJson
Introduction to websockets - Advanced methods and websocket consumption in JavaScript - Serving websockets in Python
Internet of things, introduction to Arduino and web technologies
Application exercise: "The Imitation Game" - Development of an imitating application: "Facebook", "Uber", "Open table".

Obiettivi per lo sviluppo sostenibile

Codice	Descrizione
--------	-------------