



Principali informazioni sull'insegnamento	
Denominazione dell'insegnamento	Reti di Calcolatori
Corso di studio	Informatica e Tecnologie per la Produzione del Software (ITPS)
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	Secondo
Periodo di erogazione	Primo semestre
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	Nicole Novielli
Indirizzo mail	nicole.novielli@uniba.it
Telefono	NA
Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n. 663, 6 ^a piano.
Sede virtuale	Piattaforma ADA - https://elearning.di.uniba.it/
Sito web del docente	https://collab.di.uniba.it/nicole/
Ricevimento (giorni, orari e modalità, es. su appuntamento)	Venerdì dalle 12:00 alle 14:00, previo appuntamento

Syllabus	
Obiettivi formativi	Lo studente apprenderà i concetti fondamentali delle reti di calcolatori e dei servizi e protocolli delle applicazioni di rete, con particolare riferimento alla pila di protocolli TCP/IP. Lo studente inoltre acquisirà la capacità di effettuare scelte implementative relativamente allo sviluppo di applicazioni client-server che utilizzino le socket per la comunicazione in rete. Infine, lo studente acquisirà i fondamenti della sicurezza in rete.
Prerequisiti	Si consiglia fortemente di riprendere i concetti affrontati negli insegnamenti del primo anno del corso di laurea, con particolare attenzione ai seguenti argomenti: - In riferimento all'insegnamento di Algoritmi e Strutture Dati: strutture dati, alberi, pile, dizionari, grafi, costrutti fondanti della programmazione imperativa; - In riferimento all'insegnamento di Programmazione: Programmazione in linguaggio C; - In riferimento all'insegnamento di Architetture degli Elaboratori e SO: lo stack, stratificazione a livelli, algebra di Boole e porte logiche, gestione dei processi; - In riferimento all'insegnamento di Matematica Discreta: Grafi, algebra di Boole
Contenuti di insegnamento (Programma)	<ul style="list-style-type: none">● <i>Fondamenti di reti di calcolatori</i> (9 h lezione + 6 h laboratorio)<ul style="list-style-type: none">- Introduzione alle reti di calcolatori. Definizioni base. Collegamenti diretti e indiretti. Strategie di moltiplicazione. Commutazione di circuito e di pacchetto. Caratterizzazione delle reti in base alla dimensione. Interconnessione di reti. Misure di prestazione. <ul style="list-style-type: none">- Architetture a livelli Stratificazione. Protocolli di rete. Messaggi in una pila di protocolli. Modello di riferimento ISO/OSI. Architettura di Internet. <ul style="list-style-type: none">● <i>Servizi e protocolli per applicazioni di rete</i> (9 h lezione + 9 h laboratorio)



	<ul style="list-style-type: none">- Introduzione alle applicazioni di rete <p>Definizioni base. Modelli client-server e peer-to-peer. Socket API. Scelta del tipo di servizio. Indirizzamento dei processi. Identificazione di un servizio. Concorrenza nei server. Interazione client/server orientata alla connessione e senza connessione. Collaudo di applicazioni di rete</p> <ul style="list-style-type: none">- Web <p>Introduzione. Caratteristiche fondamentali. Identificatori di risorsa: URI, URN e URL. Architettura di un browser. Architettura di un server. HTTP: formato di una richiesta e di una risposta, metodi. MIME. Trasmissione di un form. Informazioni di stato: cookies.</p> <ul style="list-style-type: none">- DNS <p>Nomi simbolici. Gerarchia dei nomi. Gerarchia dei server DNS. Risoluzione dei nomi. Caching e abbreviazioni. Descrittori di risorsa.</p> <ul style="list-style-type: none">- Programmazione delle socket <p>Programmazione delle socket in C: interazione TCP client-server con le socket; costruire messaggi; funzioni DNS; usare Socket UDP.</p> <ul style="list-style-type: none">● <i>Interconnessione delle reti</i> (10 h lezione + 15 h laboratorio) <ul style="list-style-type: none">- Trasporto in Internet <p>Il servizio di trasporto in Internet. UDP. Numeri di porta. TCP. Meccanismo di ritrasmissione. Controllo del flusso. Apertura e chiusura di una connessione. Controllo di congestione.</p> <ul style="list-style-type: none">- Instradamento in Internet <p>Obiettivo. La funzione di instradamento (routing). Costruzione della tabella di routing. Algoritmi di routing adattivi: distance vector routing e link-state routing. Routing gerarchico.</p> <ul style="list-style-type: none">- Indirizzamento in Internet <p>IPv4 datagram. Classi di indirizzi IP. Notazione decimale a punti. Indirizzi speciali e privati. Assegnazione degli indirizzi IP. Subnetting e supernetting. CIDR. Inoltro dei datagrammi IP. DHCP. IPv6. NAT.</p> <ul style="list-style-type: none">● <i>Sicurezza di rete</i> (4 h lezione) <ul style="list-style-type: none">- Attacchi alla sicurezza- Firewall- Sistemi di rilevamento delle intrusioni- Principi di crittografia <p>Crittografia a chiave simmetrica. Crittografia a chiave pubblica</p> <ul style="list-style-type: none">- Integrità dei messaggi <p>Funzioni hash crittografiche. Codice di autenticazione dei messaggi. Firme digitali</p> <ul style="list-style-type: none">- Autenticazione end-to-end- Sicurezza a livello di applicazione: email- Sicurezza a livello di trasporto: SSL- Sicurezza a livello di rete: IPsec
Testi di riferimento	<p>Teoria: J.F. Kurose & K.W. Ross, Reti di calcolatori e Internet - Un approccio top-down, Pearson</p> <p>Esercitazioni e Laboratori: M. J. Donahoo, K. L. Calvert, TCP/IP Sockets in C: Practical Guide for Programmers. Morgan Kaufman Publishers.</p>
Note ai testi di riferimento	<p>I libri di testo sono integrati con gli appunti presi a lezione e con le slide del docente disponibili sulla piattaforma di e-learning Ada.</p>
Organizzazione della didattica	
Ore	



Totali	Didattica frontale	Pratica (laboratorio, campo, esercitazione, altro)	Studio individuale
150 ore	32 ore	30 ore	88 ore
CFU/ETCS			
6 CFU	4 CFU	2 CFU	

Metodi didattici	
	Lezioni frontali supportate da slide, esercitazioni in aula, assegnazioni di esercizi di programmazione di rete con verifica in laboratorio.

Risultati di apprendimento previsti	
Conoscenza e capacità di comprensione	<i>Conoscenza e capacità di comprensione</i> Conoscere i concetti fondamentali delle reti di calcolatori. Conoscere i principali servizi e protocolli per applicazioni di rete. Comprendere i meccanismi di indirizzamento, instradamento e trasporto in Internet Conoscere i concetti fondamentali della sicurezza di rete.
Conoscenza e capacità di comprensione applicate	<i>Conoscenza e capacità di comprensione applicate</i> Acquisire familiarità con la programmazione in rete basata su interfaccia socket.
Competenze trasversali	<i>Autonomia di giudizio</i> Mostrare di aver acquisito autonomia di giudizio sulle scelte relative al funzionamento delle reti di calcolatori e dei protocolli di Internet. <i>Abilità comunicative</i> Mostrare di essere in grado di comunicare in modo appropriato le caratteristiche tecniche delle reti di calcolatori e dei protocolli di Internet <i>Capacità di apprendere in modo autonomo</i> Mostrare di aver sviluppato capacità di intraprendere in autonomia ulteriori approfondimenti su argomenti attinenti le reti di calcolatori e Internet.

Valutazione	
Modalità di verifica dell'apprendimento	L'esame si articola in due prove: <ul style="list-style-type: none">- prova di laboratorio (idoneità senza voto)- prova scritta (voto in trentesimi) <p>La <i>prova di laboratorio</i> consiste nel completare un esercizio di programmazione in C con le socket. Esempi di tali programmi possono essere reperiti nel materiale didattico distribuito dal docente sulla piattaforma di e-learning Ada, utilizzata dal Dipartimento di Informatica.</p> <p>La prova di laboratorio si svolge in uno dei laboratori didattici del dipartimento, al terzo piano e prevede una durata massima di 2 ore. Durante la prova di laboratorio, lo studente potrà utilizzare il vademecum dei prototipi delle funzioni della libreria Socket API in C messo a disposizione dal docente su Ada.</p> <p>Per gli studenti che hanno almeno il 70% di presenze del numero di ore di lezioni effettuate nel periodo precedente l'interruzione delle lezioni è prevista la possibilità di sostenere due prove intermedie di laboratorio consistenti nella consegna di due esercitazioni, una sulle Socket TCP e l'altra sulle Socket UDP. Gli studenti che</p>



	<p>consegneranno entrambe le assegnazioni in itinere, a fronte della valutazione positiva del docente potranno essere esonerati dalla prova di laboratorio. Anche in questo caso, la validità della prova di laboratorio resta pari all'anno accademico.</p> <p>La <i>prova scritta</i> consiste nel rispondere a un questionario contenente domande a risposta chiusa o aperta. Esempi di quesiti svolti sono reperibili nel materiale didattico rilasciato dal docente su Ada.</p> <p>Per sostenere la prova scritta occorre aver superato la prova di laboratorio.</p> <p>Lo studente potrà decidere di sostenere la prova di laboratorio e quella scritta in due appelli differenti. In caso di mancato superamento della prova scritta, la prova di laboratorio resta valida per l'intero anno accademico.</p>
Criteria di valutazione	<ul style="list-style-type: none">● Conoscenza e capacità di comprensione: Lo studente dovrà dimostrare di conoscere e di aver compreso i concetti fondamentali delle reti di calcolatori, i principali servizi e protocolli per applicazioni di rete, i meccanismi di indirizzamento, instradamento e trasporto in Internet, i concetti fondamentali della sicurezza di rete.● Conoscenza e capacità di comprensione applicate: Lo studente dovrà dimostrare di aver acquisito familiarità con la programmazione in rete basata su interfaccia socket. Nella fattispecie, lo studente dovrà dimostrare di essere in grado di progettare e implementare un'applicazione client-server utilizzando la Socket API in C.● Autonomia di giudizio: Lo studente dovrà dimostrare di saper formulare un proprio giudizio sulle scelte relative al funzionamento delle reti di calcolatori e dei protocolli di Internet.● Abilità comunicative: Lo studente dovrà dimostrare di saper comunicare le conoscenze acquisite nonché motivare le proprie scelte implementative in modo appropriato, con riferimento alle caratteristiche tecniche delle reti di calcolatori e dei protocolli di Internet● Capacità di apprendere: Lo studente dovrà dimostrare di aver acquisito la capacità di approfondire in autonomia gli argomenti attinenti le reti di calcolatori e Internet.
Criteria di misurazione dell'apprendimento e di attribuzione del voto finale	<p>Saranno valutati i risultati di apprendimento previsti. La prova di laboratorio è valutata in termini di idoneità a sostenere la prova scritta. La prova scritta è valutata in trentesimi.</p>
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> <p>Il materiale didattico del corso è disponibile sulla piattaforma ADA:</p> <ul style="list-style-type: none">● 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● https://www.uniba.it/it/ricerca/dipartimenti/informatica <p>Si suggerisce agli studenti di diffidare dalle informazioni circolanti su siti o gruppi social non ufficiali, poichè spesso sono risultate non affidabili, non corrette o incomplete.</p>



General information	
Academic subject	Computer Networks
Degree course	Informatica e Tecnologie per la Produzione del Software (ITPS)
Academic Year	II
European Credit Transfer and Accumulation System (ECTS)	6
Language	ING-INF/05
Academic calendar (starting and ending date)	First semester
Attendance	Not mandatory

Professor/ Lecturer	
Name and Surname	Nicole Novielli
E-mail	nicole.novielli@uniba.it
Telephone	
Department and address	Dipartimento di Informatica
Virtual headquarters	MS Teams
Tutoring (time and day)	Friday, 12:00 - 14:00

Syllabus	
Learning Objectives	The student will learn the fundamental concepts of computer networks and the services and protocols of network applications, with particular reference to the TCP / IP protocol stack. The student will also acquire the ability to make implementation choices regarding the development of client-server applications that use sockets for network communication. Finally, the student will acquire the fundamentals of network security.
Course prerequisites	Prerequisite are defined in line with the didactic regulations of the degree course, which is available on the ITPS degree course website.
Contents	<p><i>Fundamentals of computer networks</i></p> <ul style="list-style-type: none">- Introduction to computer networks. Basic definitions. Direct and indirect links. Multiplexing strategies. Circuit and packet switching. Characterization of networks based on size. Interconnection of networks. Performance measures.- Layered architectures. Stratification. Network protocols. Messages in a stack of protocols. ISO / OSI reference model. Internet architecture. <p><i>Services and protocols for network applications</i></p> <ul style="list-style-type: none">- Introduction to network applications. Basic definitions. Client-server and peer-to-peer models. Socket API. Choice of the type of service. Process addressing. Identification of a service. Competition on servers. Connection-oriented and connectionless client / server interaction. Testing of network applications- Web. Introduction. Fundamental features. Resource identifiers: URI, URN and URL. Architecture of a browser. Architecture of a server. HTTP: format of a request and a response, methods. MIME. Transmission of a form. Status information: cookies.- DNS. Symbolic names. Name hierarchy. DNS server hierarchy. Name resolution. Caching and abbreviations. Resource descriptors.- Socket programming. Socket programming in C: TCP client-server interaction with sockets; build messages; DNS functions; use UDP Socket. <p><i>Interconnection of networks</i></p> <ul style="list-style-type: none">- Internet transport. The Internet transport service. UDP. Port numbers. TCP. Retransmission mechanism. Flow control. Opening and closing a connection. Congestion control.



	<p>- Internet routing. Target. The routing function. Construction of the routing table. Adaptive routing algorithms: distance vector routing and link-state routing. Hierarchical routing.</p> <p>- Internet addressing. IPv4 datagram. Classes of IP addresses. Point decimal notation. Special and private addresses. IP address assignment. Subnetting and supernetting. CIDR. Forwarding of IP datagrams. DHCP. IPv6. NAT.</p> <p><i>Network security</i></p> <p>- Security attacks</p> <p>- Firewall</p> <p>- Intrusion detection systems</p> <p>- Principles of cryptography. Symmetric key encryption. Public key encryption</p> <p>- Integrity of messages</p> <p>Cryptographic hash functions. Message authentication code. Digital signatures</p> <p>- End-to-end authentication</p> <p>- Application level security: email</p> <p>- Transport level security: SSL</p> <p>- Network level security: IPsec</p>		
Books and bibliography	<p>Reference book:</p> <p>J.F. Kurose & K.W. Ross, Reti di calcolatori e Internet - Un approccio top-down, Pearson</p> <p>Lab on socket API programming:</p> <p>M. J. Donahoo, K. L. Calvert, TCP/IP Sockets in C: Practical Guide for Programmers. Morgan Kaufman Publishers.</p>		
Additional materials	<p>Additional book that might be used as a further reference:</p> <p>L. L. Peterson & B.S. Davie, Reti di calcolatori, Apogeo, 2008</p> <p>B. A. Forouzan, F. Mosharraf, Reti di Calcolatori - Un approccio top-down, McGraw-Hill, 2013</p> <p>Esercitazioni e Laboratori:</p> <p>B. Hall, Beej's Guide to Network Programming using Internet Sockets, Version 2.3.23, Nov. 5, 2005. http://beej.us/guide/bgnet/</p> <p>The slides are available on the course website.</p>		
Work schedule			
Total	Lectures	Hands on (Laboratory, working groups, seminars, field trips)	Out-of-class study hours/ Self-study hours
Hours			
150	32	30	88
ECTS			
4+2	4	2	
Teaching strategy	Lectures supported by slides, classroom exercises, assignments of network programming exercises with laboratory verification.		
Expected learning outcomes			
Knowledge and understanding on:	<p><i>Knowledge and understanding on:</i></p> <ul style="list-style-type: none"> - basic concepts of computer networks - main services and protocols for network applications - understanding the mechanisms of addressing, routing and transport on the Internet - fundamentals of network security. 		
Applying knowledge and understanding on:	<i>Applied knowledge and understanding</i>		



	Acquiring familiarity with socket interface-based networking programming and its use in the implementation of client-server applications.
Soft skills	<p><i>Autonomy of judgment</i> Show that you have acquired autonomy of judgment on the choices relating to the functioning of computer networks and Internet protocols.</p> <p><i>Communication skills</i> Show to be able to communicate in an appropriate way the technical characteristics of computer networks and Internet protocols</p> <p><i>Ability to learn</i> Show that you have developed the ability to independently undertake further investigations on topics relating to computer networks and the Internet.</p>
Assessment and feedback	
Methods of assessment	<p>Laboratory test and written test. The written test consists in answering a questionnaire containing closed or open questions. The lab test consists of completing a C programming exercise with sockets. To take the written test it is necessary to have passed the laboratory test. For students who have at least 70% attendance of the number of hours of lessons carried out in the period prior to the interruption of the lessons, the possibility of intermediate laboratory and written tests is foreseen.</p>
Evaluation criteria	<p><input type="checkbox"/> <i>Knowledge and understanding</i> The student will have to demonstrate knowledge and understanding of the fundamental concepts of computer networks, the main services and protocols for network applications, the mechanisms of addressing, routing and transport on the Internet, the fundamental concepts of network security.</p> <p><input type="checkbox"/> <i>Applying knowledge and understanding</i> The student will have to demonstrate that they are familiar with socket interface based network programming. In this case, the student must demonstrate that they are able to design and implement a client-server application using the C Socket API.</p> <p><input type="checkbox"/> <i>Autonomy of judgment</i> The student must demonstrate that he is able to formulate his own judgment on the choices relating to the functioning of computer networks and Internet protocols.</p> <p><input type="checkbox"/> <i>Communicating knowledge and understanding</i> The student will have to demonstrate that they are able to communicate the knowledge acquired as well as justify their implementation choices in an appropriate way, with reference to the technical characteristics of computer networks and Internet protocols</p> <p><input type="checkbox"/> <i>Capacities to continue learning</i> The student will have to demonstrate that they have acquired the ability to autonomously study topics relating to computer networks and the Internet. Expected learning outcomes will be assessed. The laboratory test is assessed in terms of eligibility to take the written test. The written test is graded out of thirty.</p>
Criteria for assessment and attribution of the final mark	Expected learning outcomes will be assessed. The laboratory test is assessed in terms of eligibility to take the written test. The written test is graded out of thirty.
Additional information	