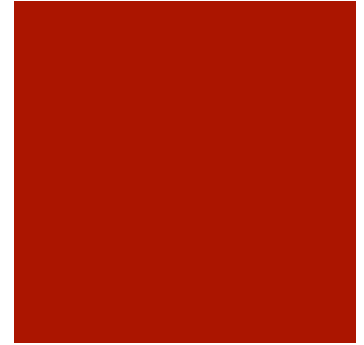




Specifiche per il progetto finale

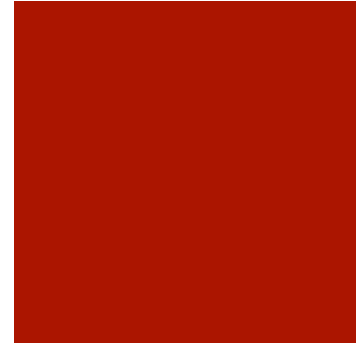
Tecnologie Web (6 cfu)
a.a. 2014-2015

Scopo del progetto



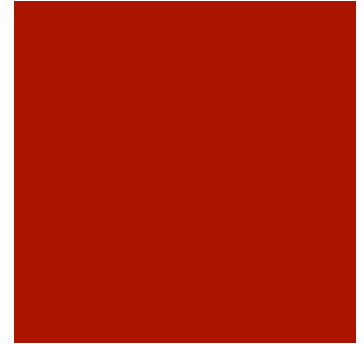
- Il progetto finale prevede la realizzazione di **un'applicazione web dinamica** incentrata su un'area tematica scelta dallo studente.
- Il progetto dovrà essere realizzato mediante le **tecnologie** proposte durante il corso e si configura come un'analisi a posteriori degli strumenti appresi attraverso i **compiti**

HTML



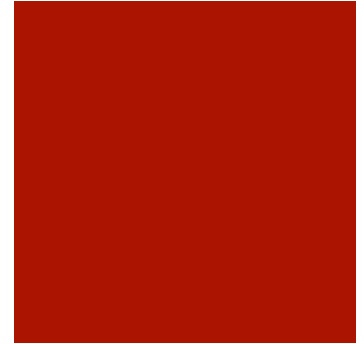
- Il progetto dovrà avvalersi delle conoscenze acquisite sul linguaggio di markup **(X)HTML**
- Tutti i principali costrutti dovranno essere presenti in varie forme all'interno del progetto (e.g., tabelle, link, immagini, form, paragrafi, titoli, etc.)
- Mantenere **rigorosamente separati la formattazione ed il contenuto** (e.g., no all'uso del tag)
- Utilizzare (dove possibile) elementi ed attributi **NON** deprecati dalla specifica W3C.
- Il codice HTML generato **dovrà** essere soggetto a **validazione** tramite il servizio della W3C (Document Type HTML 4.01 Transitional o Strict).
- L'adozione di **HTML5** è ammessa.

CSS



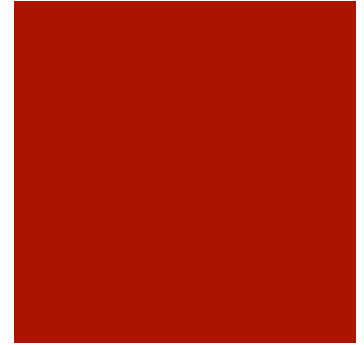
- **Separazione stile-contenuto:** il sito dovrà mostrare uno stile uniforme mediante l'utilizzo di fogli di stile esterni.
- **Posizionamento e layout tableless**
- Validazione CSS mediante il servizio W3C (<http://jigsaw.w3.org/css-validator/>)

Javascript



- **Interattività** (e.g., menù a comparsa, blocchi a scomparsa, ...)
- Gestione **eventi**
- **Modifica del DOM ed interazione con CSS**
- **Validazione** dei form, tenendo presente le contromisure atte a **prevenire attacchi XSS** e injection

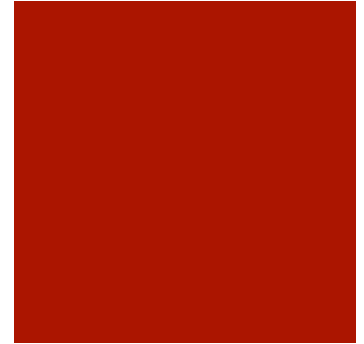
PHP



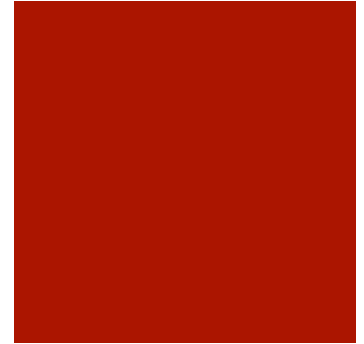
- Generazione di pagine **dinamiche ed interattive**
- **Accesso** ed **interazione** con **database** MySQL
- Gestione **cookies/sessioni**
- Nel caso in cui il browser non abbia javascript abilitato, la **validazione dati dai form** deve essere eseguita dal server tramite php
- **OPZIONALE:** Lo studente deve dimostrare di sapere usare **PHP OO!**
- **OPZIONALE:** L'architettura dell'applicazione deve seguire il pattern **MVC** (l'uso di framework non è richiesto)

Web Security

- Il progetto dovrà tenere in considerazione le linee guida illustrate relativamente alla **Web Security**.
- Il codice dovrà quindi seguire le regole base in grado di **prevenire** i principali attacchi ai quali un'applicazione web è soggetta (i.e., SQL Injection e Cross Site Scripting).

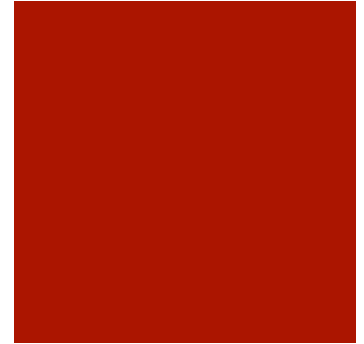


Usabilità e Design



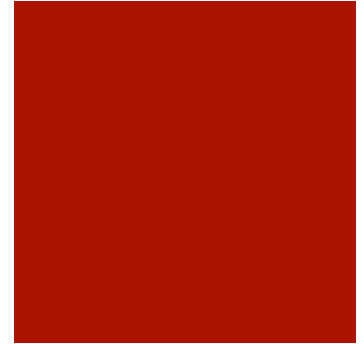
- Pur non essendo valutato interamente per le capacità di **design** ed **grafiche** (è una professionalità a parte!), è obbligatorio seguire le linee generali per la creazione di siti usabili e con un'interazione **semplice** ed **intuitiva**.
- Attenti agli accostamenti dei colori, disposizione delle informazioni, semplicità ed immediatezza nella navigazione, metafore, etc.

Ajax



- Il progetto dovrà integrare funzionalità proprie della tecnologia **Ajax**.
- Il consiglio è quello di accostarsi ad una tecnologia **molto utilizzata** nel web moderno, in ogni caso è implicito che lo studente dovrà dimostrare padronanza degli strumenti utilizzati
 - Ossia, tutto quello che scrivete nel codice è soggetto a domande!
- È possibile usare **Jquery**

Funzionalità (aggiuntive) della vostra applicazione di Social Network



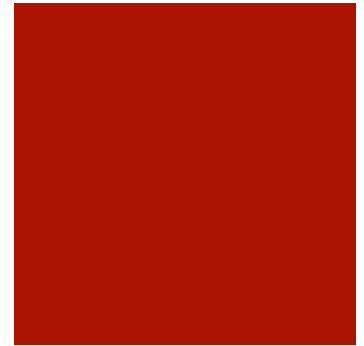
- **Registrazione utente**
- **Login utente**
- **Area privata (utente ordinario)**
- **Gestione utenti e loro contenuto (administrator)**
- **Visualizzazioni informazioni statistiche (usare Le API Google Charts)**
- **Ricerca nuovi amici**

Proprietà “ereditate” dai compiti



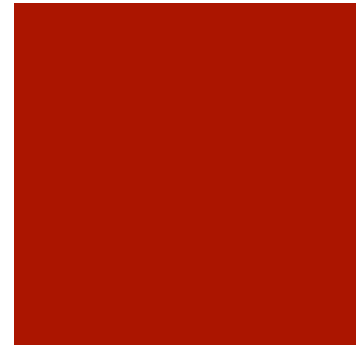
- Una volta effettuato il login, ogni utente ha a disposizione queste sezioni:
 - Home (profilo), Amici, Notizie, Statistiche
 - La navigazione tra le sezioni (e sottosezioni eventuali) deve essere facilitata da opportuni menù di navigazione verticali (e/o orizzontali), seguendo un layout a doppia o tripla colonna – con header e footer
- Per gli studenti **non frequentanti**: vedere i testi dei compiti per maggiori informazioni

Inoltre



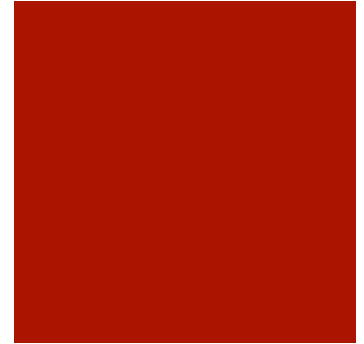
- La sezione **notizie** diventa "viva" e si aggiorna al cambiamento di stato degli amici. I contenuti degli amici devono apparire in questa pagina in ordine cronologico.
- sezione **statistiche** diventa "viva" in base ai dati reali e grazie alle api google charts
- sezione contattami viene sostituita da "**cerca amico**", per trovare altri utenti e chiedere loro il contatto

Utenti



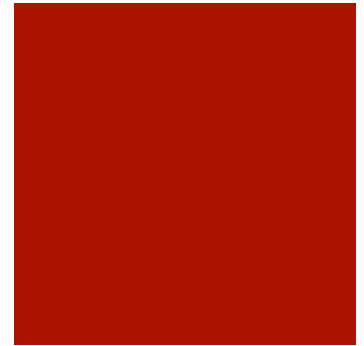
- **Utente ordinario:** utente standard che accede al sito previa registrazione. La registrazione ha effetto immediato e l'utente sarà abilitato all'uso del sito da subito, senza che la sua registrazione sia abilitata dall'amministratore
- **Administrator:** utente con privilegi di amministrazione. Questo utente non può essere rimosso e deve essere in grado di rimuovere e/o modificare i contributi di tutti gli utenti.

Registrazione utente



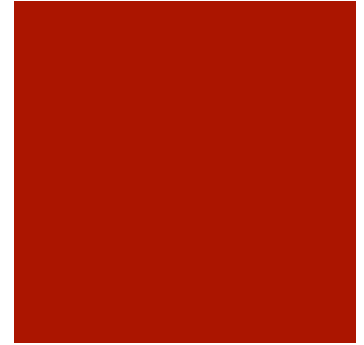
- Implementare le funzionalità di **registrazione** di un nuovo utente. La registrazione prevede i seguenti passi:
 - Invio di un modulo da parte dell'utente con i dati personali **privati** (username, password, generalità, email, etc.).
 - I dati da mostrare nel **profilo pubblico** (nome, cognome, data di nascita, fotografia personale) possono essere inseriti solo dopo che l'account sarà attivato, senza appesantire la procedura di registrazione

Area privata (utente ordinario)



- Un utente ordinario, dopo aver eseguito la procedura di login, avrà la possibilità di inserire/modificare i dati di profilo mediante l'accesso ad un'area privata.
- Deve essere possibile:
 - Modificare dati utente.
 - Modificare la password (ma non lo username).

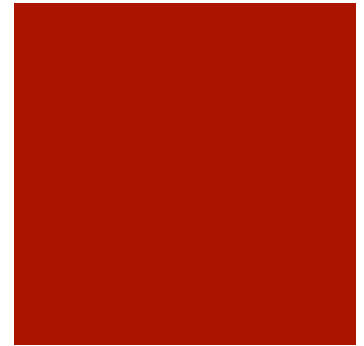
Gestione account utenti



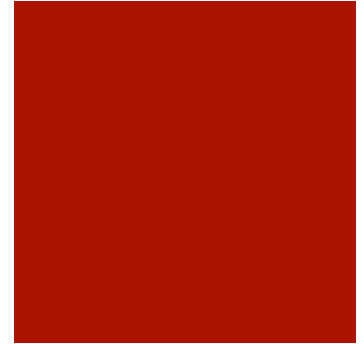
- Funzionalità disponibile solo per l'utente administrator. Se il login è effettuato con le credenziali di amministratore, si deve avere la possibilità di:
 - Visionare tutti gli utenti iscritti
 - Eliminare utenti
 - Eliminare/modificare contenuti degli utenti

Login utente

- Il sito deve implementare un meccanismo di **login** tramite la specifica di credenziali utente nella forma di **username** e **password**.
- A login effettuato, una sessione utente deve essere creata o ripristinata.
- Tale sessione dovrà contenere lo stato dell'interazione utente-sistema

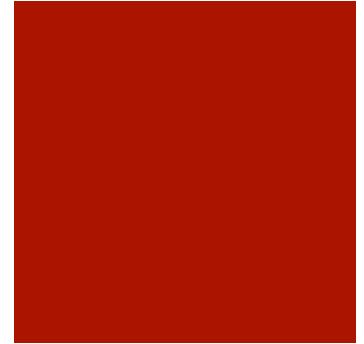


Inserimento e modifica dati



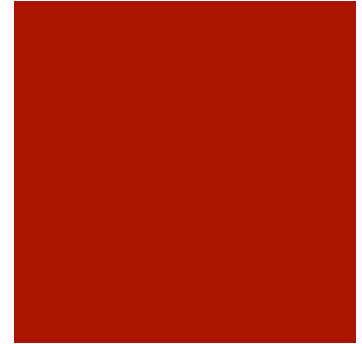
- Il servizio dovrà fornire le funzionalità di:
 - **Aggiornamento** del proprio messaggio di stato (da adesso in poi il messaggio di stato di un utente è considerato un **contenuto**)
 - **OPZIONALE:** ad un nuovo contenuto inserito si può associare un'immagine
 - **Modifica** e **cancellazione** contenuti
 - **Commentare** i contenuti o il messaggio di stato dei propri amici
 - Tutti i dati devono essere **validati** a livello **js** (ove possibile) e a livello **php**

Inserimento e modifica dati



- Tali funzionalità sono strettamente legate alla struttura del database.
- Ricordo quindi che la fase di design e progettazione della base di dati è di fondamentale importanza e sarà argomento di discussione in fase di esame.

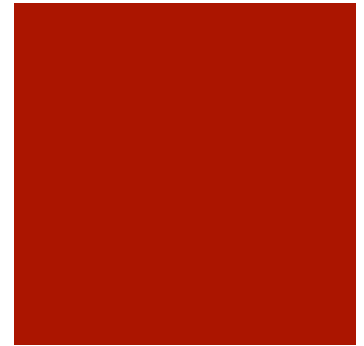
Amici: ricerca e visualizzazione profili



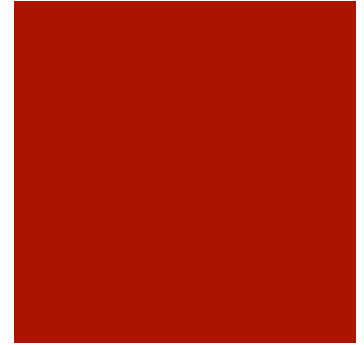
- L'utente deve poter **cercare** tra gli utenti suoi potenziali contatti e **visualizzare** i profili generici (le "home") degli amici
 - Perché un utente diventi **amico**, deve essere fatta esplicita richiesta. Questa deve quindi essere **notificata** ed infine **reciprocata**. Solo in quel caso l'amicizia sarà considerata **stabilita**
 - Solo quando l'amicizia è stabilita, i profili degli utenti saranno reciprocamente **visualizzabili**
- Definire criteri di ricerca
- L'utente dovrà inserire i parametri di ricerca tramite form o controlli specifici (campi input, menù a tendina, etc.)
- Tutti i parametri in input dovranno essere validati.

Risultati della Ricerca

- Il sistema dovrà fornire la possibilità di ordinare i risultati provenienti da una ricerca secondo criteri scelti dall'utente.
 - Es.: ordinamento alfabetico, per data o per luogo di residenza

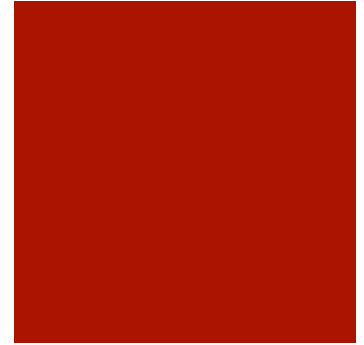


Funzionalità opzionali (obbligatorie se NON frequentanti)



- **Ricerca avanzata dei contenuti:** si possono fornire alcuni parametri per ricerche specifiche; ad esempio, si vogliono vedere i contenuti inseriti dagli utenti con maggior numero di “like”, oppure i messaggi contenenti una data parola o hashtag. Una volta restituita la lista dei contenuti corrispondenti alla ricerca, devono essere presenti anche i link agli utenti che li hanno prodotti.
- **Notifiche:** Ogni volta che un utente inserisce un nuovo contenuto e/o effettua un commento, gli amici devono ricevere una notifica.

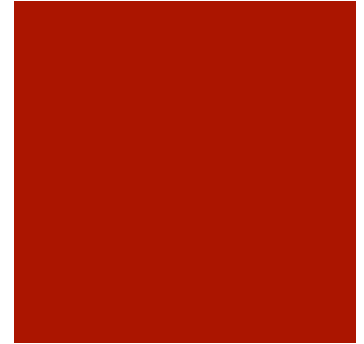
Funzionalità opzionali (obbligatorie se NON frequentanti) - 2



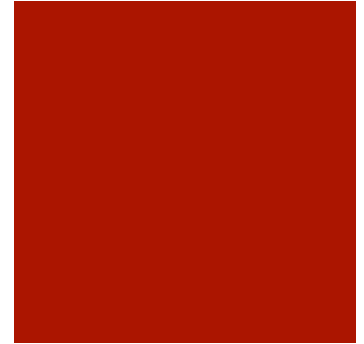
- **Attivazione utenti:** dopo la registrazione, l'utente deve attendere che l'amministratore abiliti il suo account. Prima di questo passo, non sarà possibile accedere ai servizi riservati
- **Opzionale o non opzionale?** Tutto ciò che è indicato come opzionale nelle specifiche di questo progetto è da considerarsi **obbligatorio** per i non frequentanti.

Estensioni/variazioni

- Essendo un progetto aperto, è lasciata allo studente la possibilità di personalizzazione le funzionalità implementate.
- Esempio di variazione: le amicizie possono essere stabilite alla “Twitter”. Possiamo decidere di **seguire** gli aggiornamenti di alcuni utenti, anche se quelli non hanno reciprocato. Però deve essere chiaro a quel punto chi segue chi.
- In ogni caso, si consiglia di discutere con il docente idee, proposte e soluzioni.
- **Consiglio:** non siate troppo ambiziosi, non vogliamo TUTTE le funzionalità di Facebook... I Social Media oggi popolari non sono stati realizzati in una notte (e neanche in un mese...)

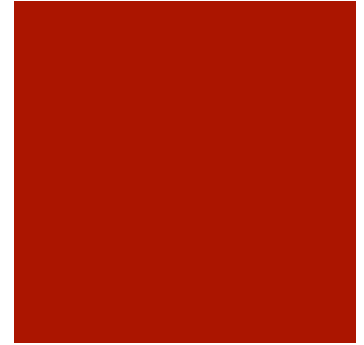


Modalità di esame - 1



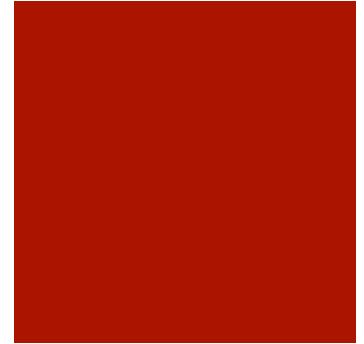
- Il progetto deve essere svolto in modalità **individuale**.
- Solo per gli studenti **FREQUENTANTI**: Prima della prova orale **sarà assegnato un voto al progetto che deve essere confermato**. Dopo la conferma, si calcolerà la media pesata del progetto (che vale il 60% del voto finale). Il risultato sarà registrato come voto d'esame.
- Il progetto può essere valutato come non idoneo o non corrispondente alle specifiche o privo di adeguate spiegazioni in fase di orale, il che significa esame **NON** superato.

Modalità di esame - 2



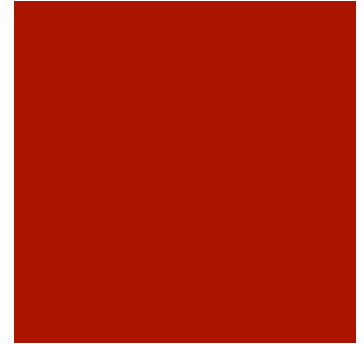
- È possibile discutere il laboratorio solo se l'elaborato è stato consegnato almeno **3 giorni prima** dell'appello.
- Gli elaborati non consegnati per tempo **non** verranno corretti e lo studente potrà presentarsi solo all'appello immediatamente successivo.
- La discussione del laboratorio sarà completata da **domande sul codice** nonché da **domande teoriche correlate**.

Modalità di esame - 4



- L'elaborato deve essere perfettamente funzionante nelle macchine di laboratorio e dovrà essere **installato entro tre giorni prima** della discussione orale.
- **Consegna:** Insieme all'installazione del progetto, dovete consegnare una **relazione finale**, contenente una sezione **HOWTO** che spieghi esattamente come usare il progetto. In particolare sono necessarie le seguenti informazioni:
 - **URL** del vostro progetto:
`http://labappweb.labinfo.educ.di.unito.it/studenti/[LOGIN UTENTE]/[CARTELLA PROGETTO]`
 - **Username/password** dell'utente **amministratore**
 - **Username/password** degli utenti già **registrati**
- La **relazione** deve fornire una descrizione di massima delle soluzioni adottate e delle funzionalità implementate
- Il tutto non deve superare le **due pagine** di lunghezza.

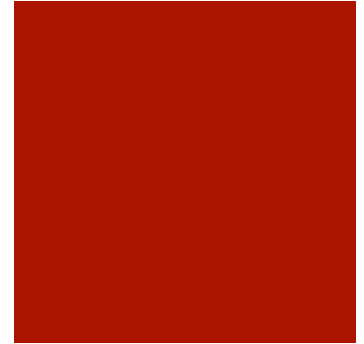
Modalità di esame - 5



- Per fare un esempio, qualsiasi comando mySQL (anche per creare le tabelle) dovrà essere eseguito dal codice e **l'esaminatore non dovrà fare alcun lavoro extra** per capire come fare funzionare il sito!
- Se si deve fare del lavoro extra: il file HOWTO deve spiegare cosa fare
- La non corretta osservanza di questo punto comporterà una **penalità**.

Modalità di esame - 6

- Qualsiasi deroga alle linee guida delineate in questo documento **dovranno essere preventivamente discusse con il docente** previa la non validità del progetto presentato.



Note per gli studenti NON frequentanti



- Studente **frequentante**: chi ha consegnato **almeno due compiti** su quattro nell'anno accademico in corso.
- Studente **non frequentante**: **tutti gli altri**.
- **Consiglio**: se avete consegnato **al massimo due compiti** su quattro, sarete molto penalizzati anche consegnando un buon progetto. Meglio presentarsi come **non frequentanti**, dando comunicazione al docente contestualmente alla consegna della relazione finale.

Note per gli studenti NON frequentanti - 2



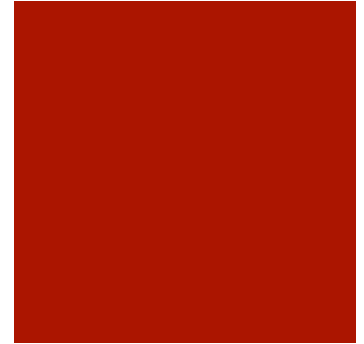
- Le **scadenze** sono identiche a quelle per gli studenti frequentanti.
- La valutazione al progetto sarà assegnata solo dopo **l'esame orale**, che ha lo scopo di analizzare nel dettaglio l'intera struttura applicativa.
- Il **voto finale** sarà registrato solo dopo il superamento della discussione del progetto.

Note per gli studenti NON frequentanti - 3



- Dovranno essere implementate **obbligatoriamente** anche le funzionalità lasciate come opzionali per gli altri studenti (vedi diapositiva **n. 20**)

Riepilogo



- **Consegna elaborato:**

- Entro **tre** giorni prima della data dell'appello scelto, entro le ore 23:55 (via moodle)
- Ci sarà sempre una cartella pronta a ricevere i vostri elaborati prima dell'appello

- **DISCUSSIONE PROGETTO FINALE:**

- Guardate le date degli appelli ufficiali
- Eventuali modifiche alle date e all'orario verranno comunicate sul **FORUM** del corso.
- Per la discussione **è necessario iscriversi** tramite la procedura elettronica all'appello corrispondente.