

Laboratorio di Informatica

Presentazione del Corso

docente: Cataldo Musto

cataldo.musto@uniba.it

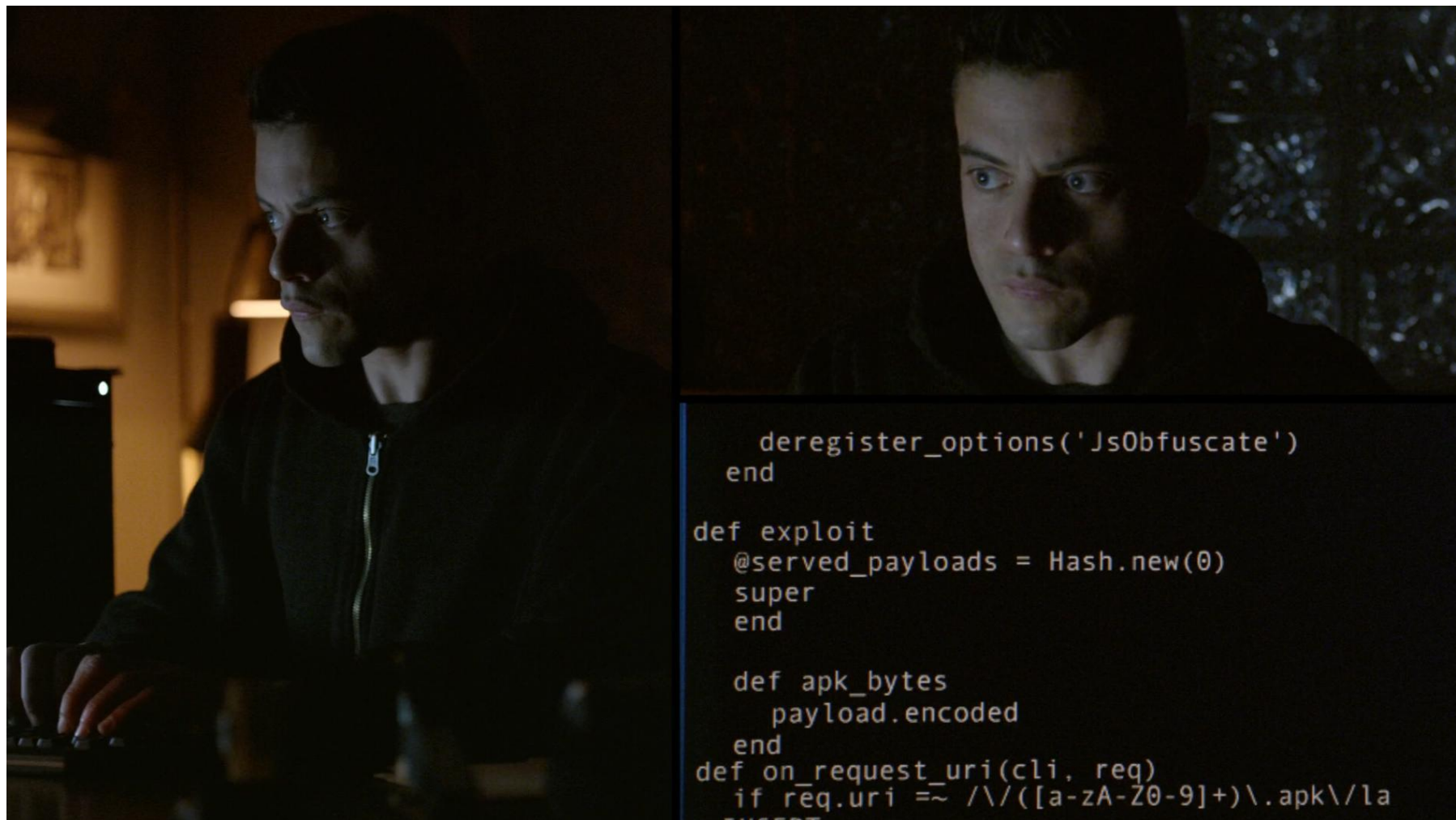
Dove mi trovo? E' il posto giusto per me?

Dove mi trovo? E' il posto giusto per me?

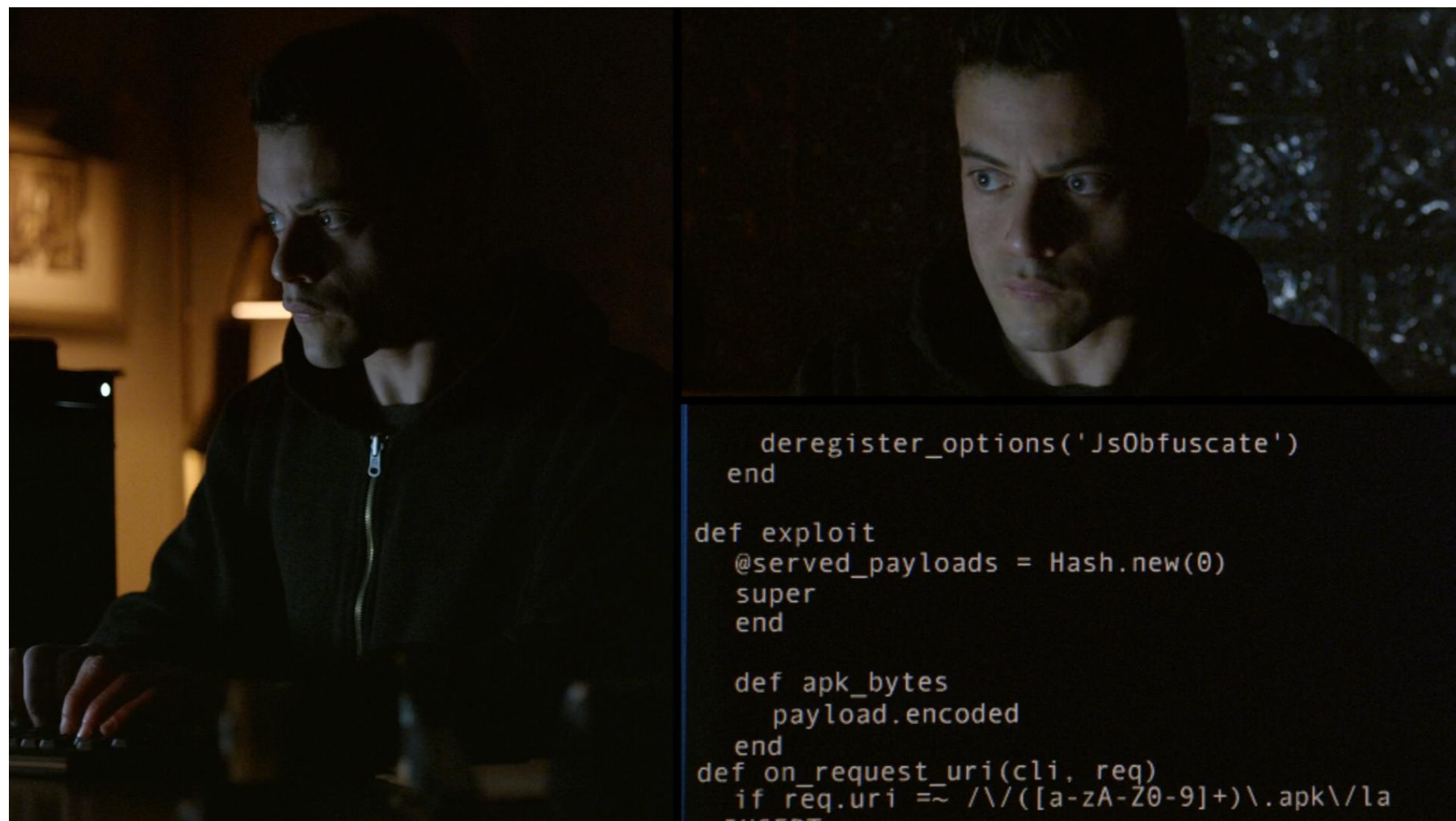
- E' **importante** aver seguito e studiato (e magari sostenuto)
 - Programmazione
- E' **consigliabile** aver seguito e studiato
 - Architettura degli Elaboratori
 - Sistemi Operativi
- Il **Laboratorio di Informatica è propedeutico** agli insegnamenti nei settori INF/01 e ING-INF/05 del secondo anno (ovvero del terzo e quarto anno per gli studenti non impegnati a tempo pieno)
 - **Fonte:** <https://goo.gl/rbTA96> (pagina 14)

Laboratorio di Informatica: a che serve?

Laboratorio di Informatica: a che serve?



Laboratorio di Informatica: a che serve?



Quasi.

Cosa insegna?

- **Aspetti pragmatici della programmazione**
 - Pragmatico=?

Cosa insegna?

- **Aspetti pragmatici della programmazione**
 - Pragmatico: che riguarda **l'attività pratica** (in greco, *πραγμα* significa «cosa»)

Cosa insegna?

- **Aspetti pragmatici della programmazione**
 - Pragmatico: che riguarda **l'attività pratica** (in greco, *πραγμα* significa «cosa»)
- **Scrivere programmi che risolvano correttamente i problemi**
 - Programmazione «difensiva»
 - Programmazione modulare
 - Testing del codice
 - Debugging del codice
 - Documentazione del codice
 - Stile di programmazione

Programmazione «difensiva» ?

Programmazione «difensiva»

- Scrivere del codice che sia in grado di prevenire o gestire **le situazioni di errore prevedendo e individuando i casi «limite»** che possono caratterizzare il comportamento di un algoritmo
- **Esempio**
 - Scrivere un algoritmo che a seconda dell'età inserita dall'utente stampi un messaggio se l'individuo è minorenne

Programmazione «difensiva»

- Scrivere del codice che sia in grado di prevenire o gestire **le situazioni di errore prevedendo e individuando i casi «limite»** che possono caratterizzare il comportamento di un algoritmo
- **Esempio**
 - Scrivere un algoritmo che a seconda dell'età inserita dall'utente stampi un messaggio se l'individuo è minorenne

```
int age = 0;  
printf(«%s», «Inserisci la tua età»);  
scanf(«%d», &age)
```

Programmazione «difensiva»

- Scrivere del codice che sia in grado di prevenire o gestire **le situazioni di errore prevedendo e individuando i casi «limite»** che possono caratterizzare il comportamento di un algoritmo
- **Esempio**
 - Scrivere un algoritmo che a seconda dell'età inserita dall'utente stampi un messaggio se l'individuo è minorenne

```
int age = 0;
printf(«%s», «Inserisci la tua età»);
scanf(«%d», &age)

if(age < 18) {
    printf(«%s», «utente minorenne»);
}
```

Programmazione «difensiva»

- Scrivere del codice che sia in grado di prevenire o gestire **le situazioni di errore prevedendo e individuando i casi «limite»** che possono caratterizzare il comportamento di un algoritmo

**Cosa succede se l'utente inserisce un valore non realistico?
Es. 400 oppure -20 oppure scrive DICIOTTO**

```
int age = 0;
printf(«%s», «Inserisci la tua età»);
scanf(«%d», &age)

if(age < 18) {
    printf(«%s», «utente minorenne»);
}
```

Programmazione «difensiva»

- Scrivere del codice che sia in grado di prevenire o gestire **le situazioni di errore prevedendo e individuando i casi «limite»** che possono caratterizzare il comportamento di un algoritmo
- **Esempio**
 - Scrivere un algoritmo che a seconda dell'età inserita dall'utente stampi un messaggio se l'individuo è minorenne

```
int age = 0;  
printf(«%s», «Inserisci la tua età»);  
scanf(«%d», &age)
```

```
if(age < 0) {  
    printf(«%s», «valore non valido»);  
}
```

```
else if(age < 18) {  
    printf(«%s», «utente minorenne»);  
}
```

```
else if(age > 120) {  
    printf(«%s», «valore non valido»);  
}
```

**Esempio di
Programmazione
Difensiva**

Programmazione modulare

- Un errore comune è quello di strutturare i programmi all'interno di un unico file sorgente e in un unico metodo
- **La programmazione modulare suggerisce di «suddividere» il codice sorgente in unità più piccole**
 - Funzioni (che possono essere richiamate in punti diverse del programma)

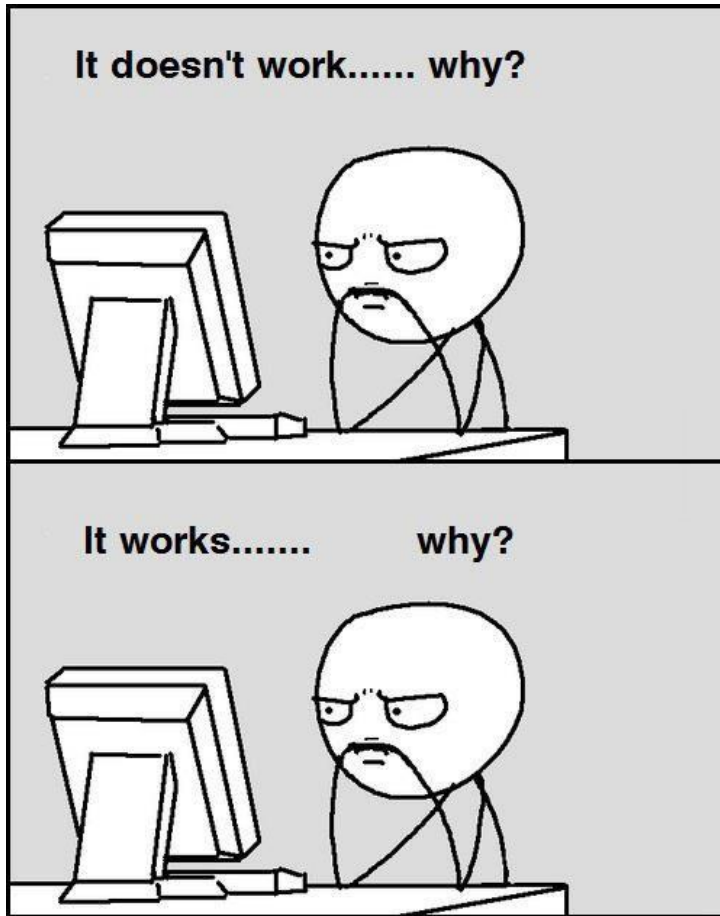
Programmazione modulare

- Un errore comune è quello di strutturare i programmi all'interno di un unico file sorgente e in un unico metodo
- **La programmazione modulare suggerisce di «suddividere» il codice sorgente in unità più piccole**
 - Funzioni (che possono essere richiamate in punti diverse del programma)
- **Uno dei principi della programmazione modulare è quello di aggregare un insieme di funzioni in particolari tipi di file detti «librerie»**
- Possono essere riutilizzati in programmi diversi (o anche da persone diverse)
 - **Esempio:** ho un metodo che verifica che il *codice fiscale inserito da un individuo* sia corretto. Metodi che possono essere riutilizzati vengono tipicamente inseriti nelle librerie!

Programmazione modulare

- Un errore comune è quello di strutturare i programmi all'interno di un unico file sorgente e in un unico metodo
- **La programmazione modulare suggerisce di «suddividere» il codice sorgente in unità più piccole**
 - Funzioni (che possono essere richiamate in punti diverse del programma)
- **Uno dei principi della programmazione modulare è quello di aggregare un insieme di funzioni in particolari tipi di file detti «librerie»**
- Possono essere riutilizzati in programmi diversi (o anche da persone diverse)
 - **Esempio:** ho un metodo che verifica che il *codice fiscale inserito da un individuo* sia corretto. Metodi che possono essere riutilizzati vengono tipicamente inseriti nelle librerie!
 - Ogni qual volta si inseriscono dei file .h nel codice in realtà si sta importando una libreria (**#include <stdio.h>**)

Testing, Debugging e Documentazione

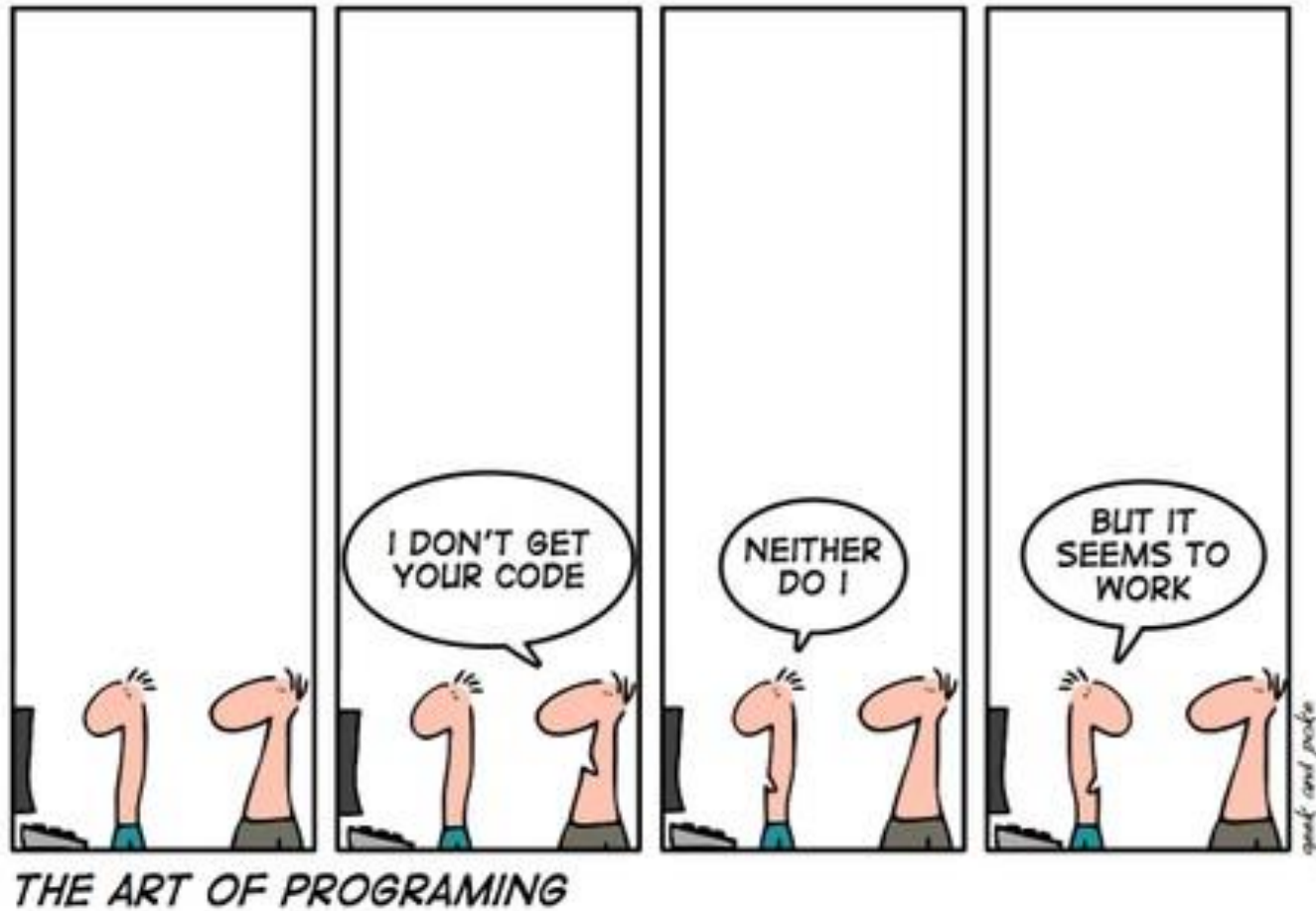


Testing, Debugging e Documentazione

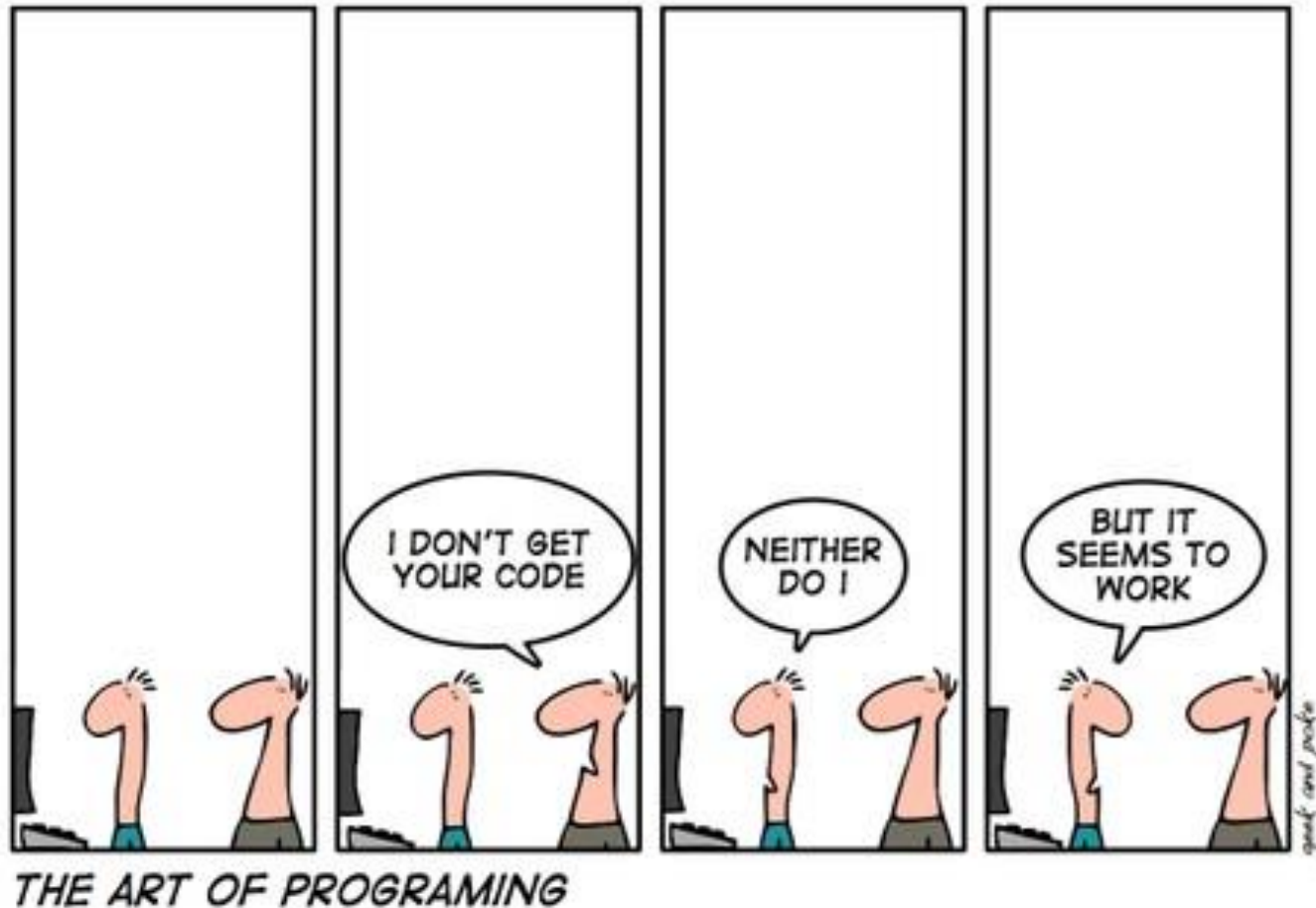


- Acquisire consapevolezza nell'uso degli **strumenti** per
 - Individuare (e rimuovere) **bug** presenti nel codice
 - **Definire dei «casi di test»** che verifichino il corretto funzionamento dei programmi
 - Generare in **automatico la documentazione** del codice (a partire dai commenti)

Stile di Programmazione



Stile di Programmazione



- Acquisire le metodologie corrette per
 - Assegnare **nomi significativi alle variabili**
 - **Indentare** correttamente e **commentare** il codice sorgente
- Utilizzare uno stile uniforme che faciliti la comprensione del **codice sorgente**

Obiettivi professionalizzanti: cosa imparo?

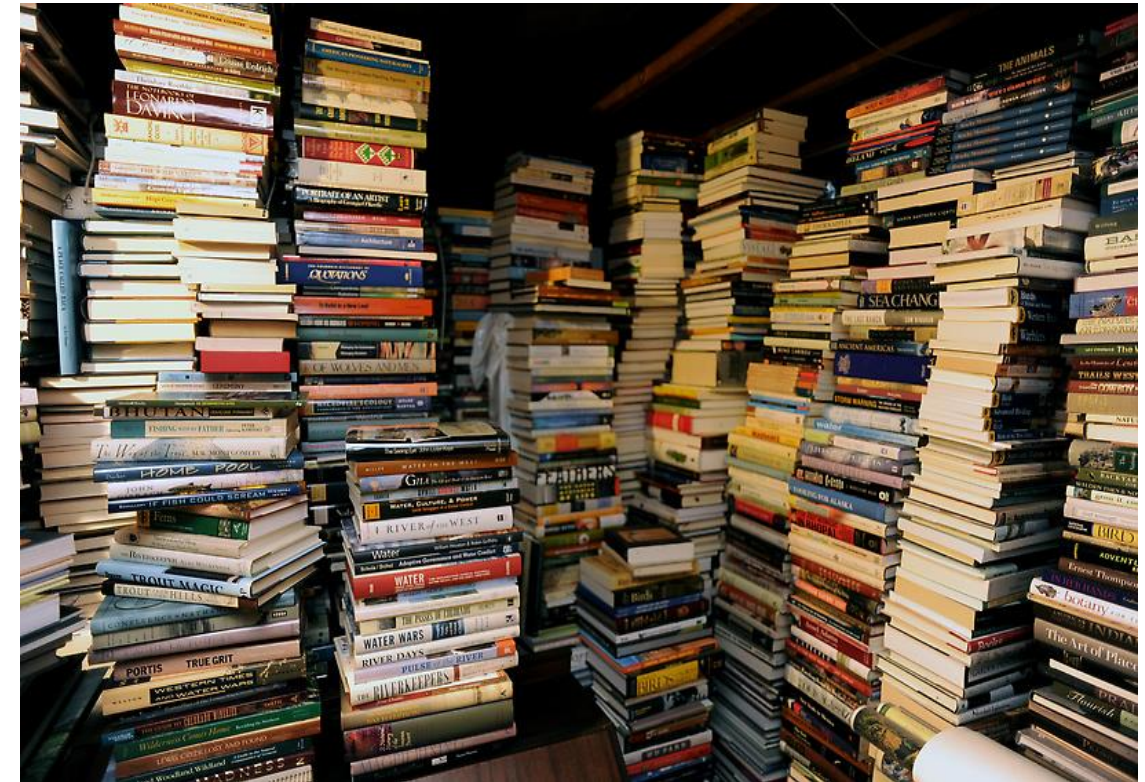
- Saper scrivere **programmi correttamente funzionanti** e ben documentati
- **Strutturare correttamente il codice sorgente** in moduli e librerie, applicando **i principi dell'astrazione funzionale**
- **Saper verificare empiricamente la correttezza dei programmi** mediante testing
- Saper **individuare (e risolvere) malfunzionamenti** attraverso il debugging
- Ampliare le **capacità di problem-solving**

Obiettivi professionalizzanti: cosa imparo?

Imparare a programmare: meglio

Da che materiale posso studiare?

- **Dispense fornite dal docente**
- **Testi consigliati**
 - B.W. Kerninghan, R. Pike, *"Programmazione nella pratica"*, Addison Wesley, 1999
 - Jeri R. Hanly, Elliot B. Koffman, *"Problem solving e programmazione in C"*, Apogeo, 2013
- **Testo di riferimento per il Linguaggio C**
 - Deitel P. & H., *"Il linguaggio C - Fondamenti e tecniche di programmazione"*, ed. marzo 2013 - Pearson (ISBN: 9 788871 929378)



Come è strutturato il corso?

- **6 CFU = 6 * 25 ore di impegno studente**
 - 3 CFU T1 (lezione) = 24 ore di lezione frontale
 - 3 CFU T2 (laboratorio) = 45 ore di esercitazioni
- **Lezioni**
 - Mercoledì 8.30 – 12.30
 - Giovedì 11.30 – 14.30



Struttura del Corso – Lezioni Frontali

- In Aula Magna
- **Frequenza Consigliata**
- Si affrontano gli **aspetti teorici** del corso
 - *Come si usa un debugger? Cosa è il testing? Come si definiscono i casi limiti di un programma? Come si genera della «buona» documentazione? Quali sono i principi della programmazione modulare?*
- Ogni lezione
 - **Illustra i contenuti teorici**
 - **Li collega ad esercizi pratici da sviluppare in Laboratorio**

Struttura del Corso – Lezioni di Laboratorio

- **Sempre in Aula Magna**

- Portate i vostri notebook (carichi!) , l'Aula è elettrificata
- **Valuteremo in seguito l'utilizzo dei Laboratori**

- Massimo due persone per notebook (meglio una)

- Lo scopo è di svolgere l'esercitazione assegnata
 - Le esercitazioni saranno erogate sulla piattaforma **Repl.it** (<http://repl.it>)
- Completamento a casa
 - 1 CFU = 15 ore in aula + **10 ore a casa**

Struttura del Corso – Lezioni di Laboratorio

- **Nota molto molto molto molto ma molto molto importante**
 - Il docente non è un **debugger**
 - Il docente non è un **tester**
 - **Il docente non è un conoscitore di tutti i possibili problemi che possono essere causati dall'uso errato del Linguaggio C 😊**

Programma del Corso

- Revisione/Approfondimento Linguaggio C
 - Richieste?
- Programmazione Modulare
- Stili di programmazione
- Testing e Debugging
- Documentazione del codice
- Algoritmi Fondamentali
 - Ricerca & Ordinamento



Ulteriori dettagli su: <http://informatica.uniba.it>

Link al programma: <http://tinyurl.com/programma-labinf1718>

Come possiamo interagire?

- **Quesiti di interesse comune?**

- Attraverso la piattaforma Ada
(<https://elearning.di.uniba.it/course/view.php?id=131>)
- Il materiale del corso sarà condiviso sulla piattaforma
- Non è un help-desk h/24 ;)
- Gli studenti possono (devono!) collaborare

- **Quesiti personali?**

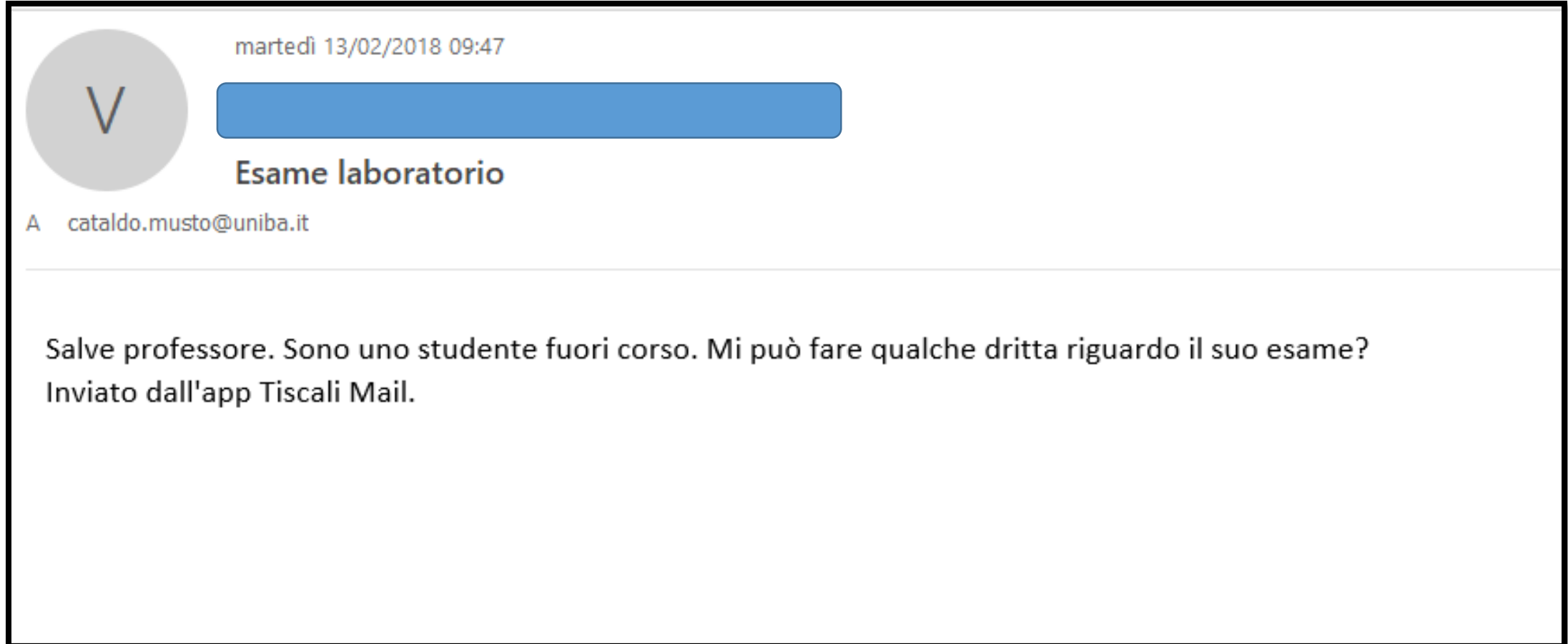
- cataldo.musto@uniba.it

- **Non vi basta?**

- Ricevimento **Giovedì 15.00-17.00 dopo la lezione**
- Concordare prima (preferibilmente)



Esempi di mail da non mandare



Esempi di mail da non mandare



Utilizzo di Ada

(<https://elearning.di.uniba.it/course/view.php?id=131>)

**UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO**

Laboratorio di Informatica (co ...



 Home  Dashboard  Eventi  i miei corsi  Nel corso

 Attiva modifica  Nascondi blocchi  Standard view

 > I miei corsi > Laboratorio di Informatica (corso B)

Laboratorio di Informatica (corso B)

a.a. 2017/2018

Prof. Cataldo Musto

CdS in Informatica e Tecnologie per la Produzione del Software(ITPS)
II anno, I semestre, 6 CFU (3+3)

Orario delle lezioni

corso B (lettere M-Z., Aula Magna)

Mercoledì ore 08.30 - 12.30
Giovedì ore 11.30 - 14.30

Calendario



febbraio 2018

Dom	Lun	Mar	Mer	Gio	Ven	Sab
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			

-  Nascondi eventi globali
-  Nascondi eventi di categoria
-  Nascondi eventi di corso
-  Nascondi eventi di gruppo
-  Nascondi eventi dell'utente

Utilizzo di Ada

Password: labinf-1718

(<https://elearning.di.uniba.it/course/view.php?id=131>)

**UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO**

Laboratorio di Informatica (co ...



 Home  Dashboard  Eventi  i miei corsi  Nel corso

 Attiva modifica  Nascondi blocchi  Standard view

 > I miei corsi > Laboratorio di Informatica (corso B)

Laboratorio di Informatica (corso B)

a.a. 2017/2018

Prof. Cataldo Musto

CdS in Informatica e Tecnologie per la Produzione del Software(ITPS)
II anno, I semestre, 6 CFU (3+3)

Orario delle lezioni

corso B (lettere M-Z., Aula Magna)

Mercoledì ore 08.30 - 12.30
Giovedì ore 11.30 - 14.30

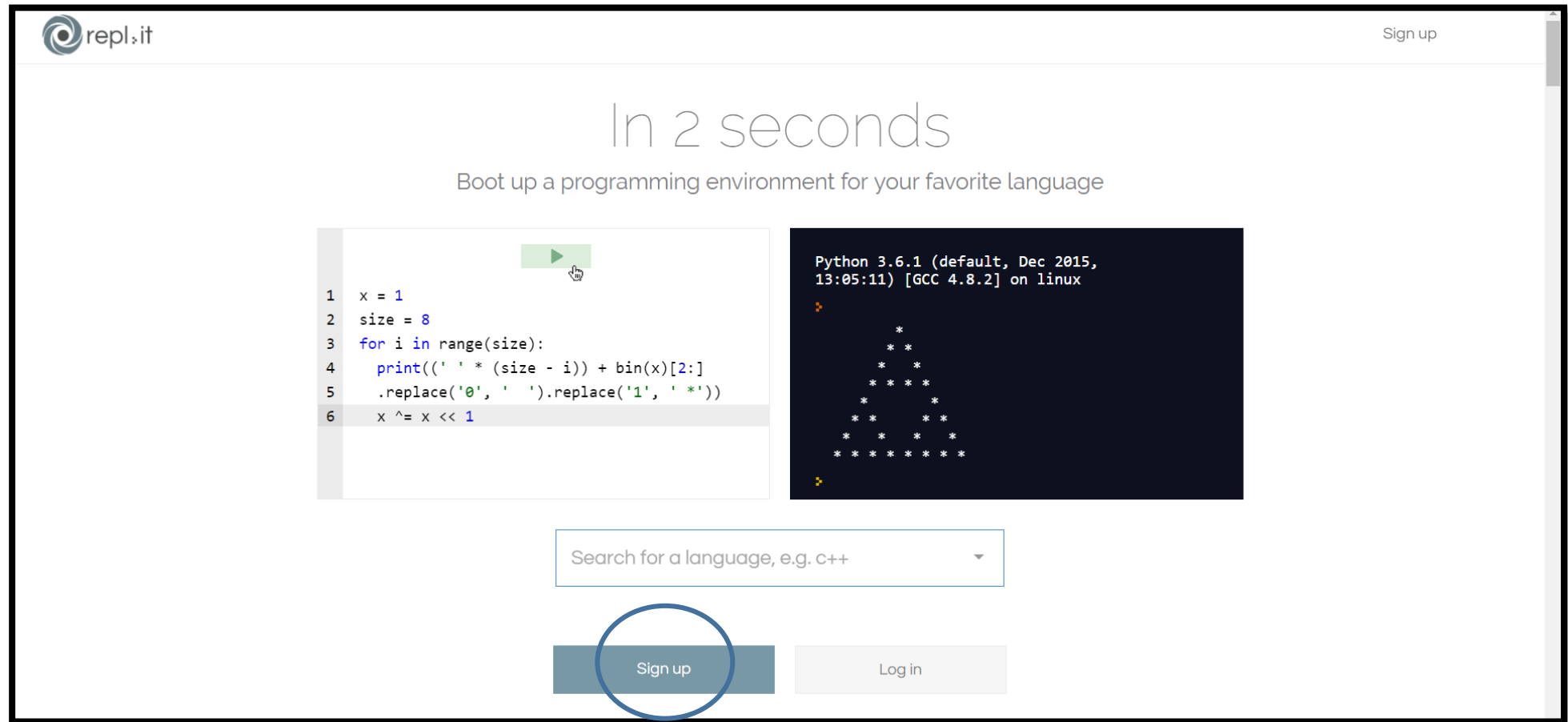
Calendario

febbraio 2018

Dom	Lun	Mar	Mer	Gio	Ven	Sab
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			

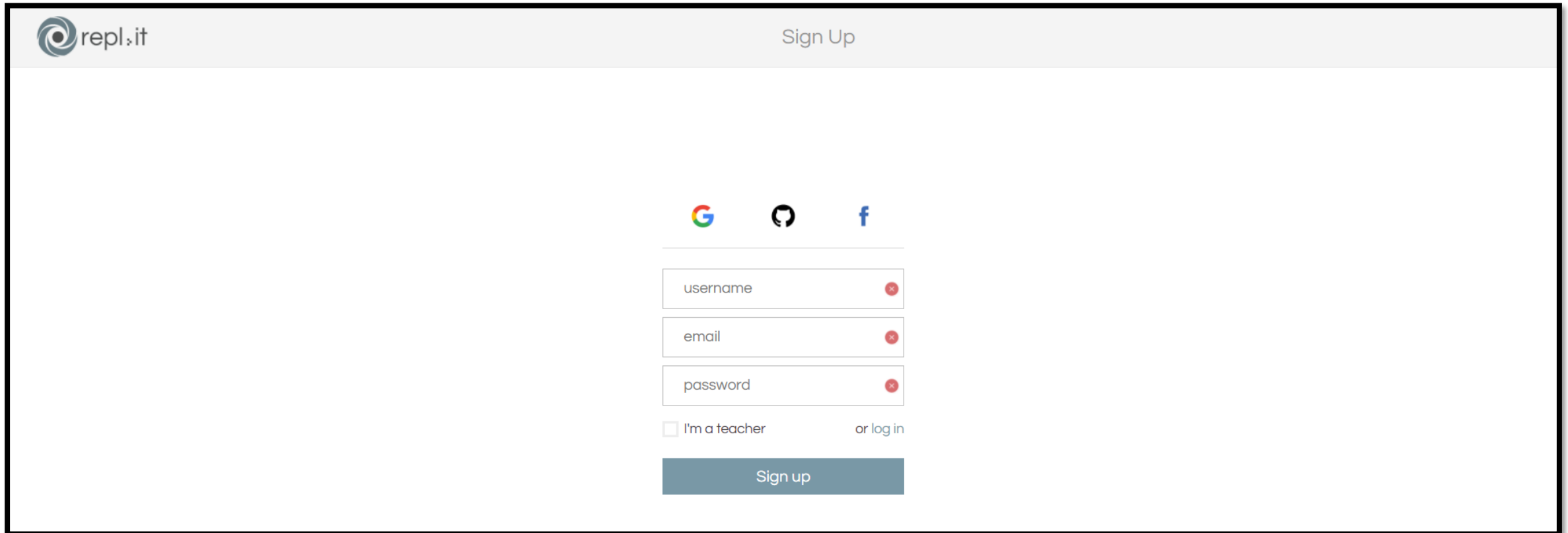
-  Nascondi eventi globali
-  Nascondi eventi di categoria
-  Nascondi eventi di corso
-  Nascondi eventi di gruppo
-  Nascondi eventi dell'utente

Utilizzo di Repl (<http://repl.it>)



Iscrizione alla piattaforma

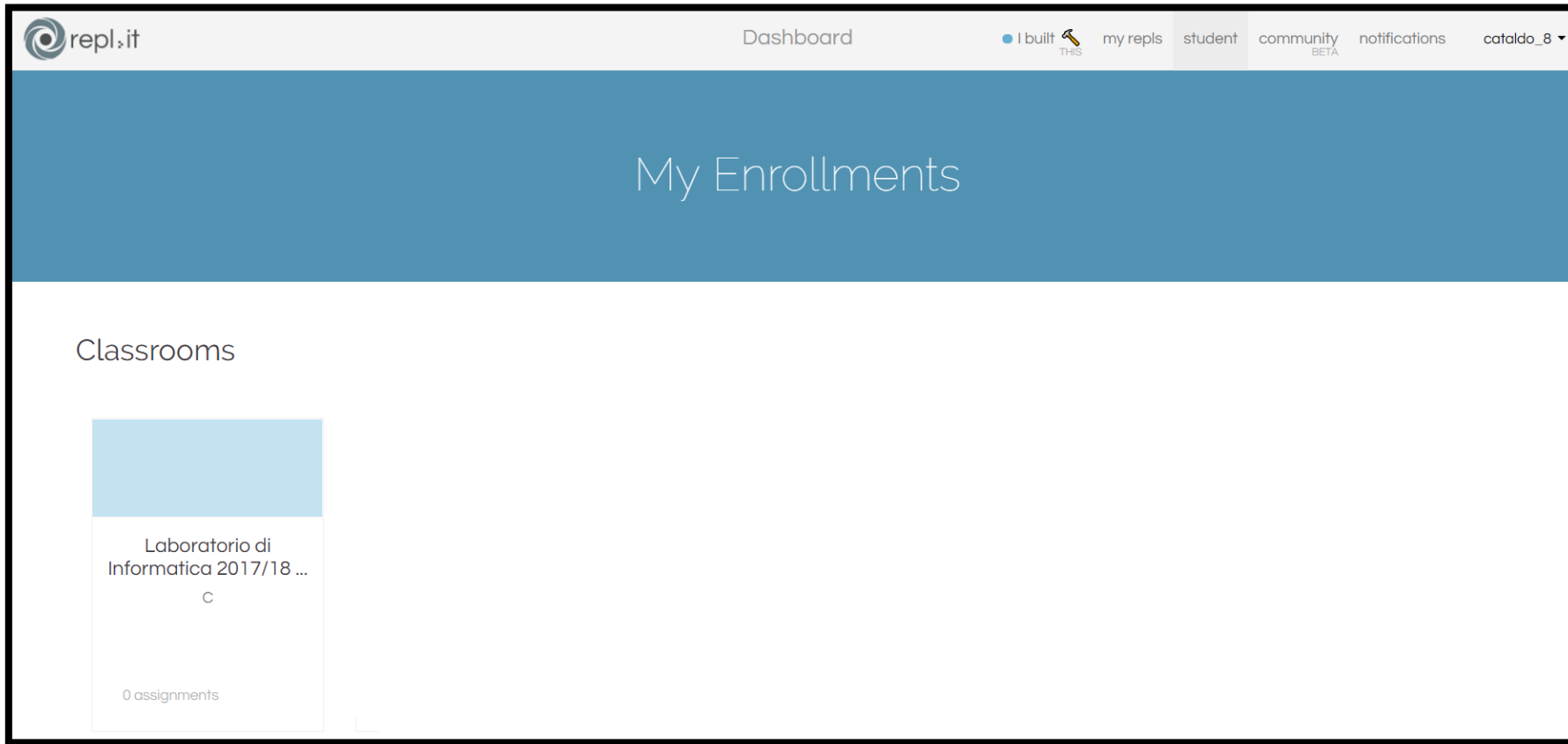
Utilizzo di Repl (<http://repl.it>)



The screenshot shows the Repl.it sign-up interface. At the top left is the Repl.it logo, and at the top right is a 'Sign Up' link. Below these are three social media icons: Google, GitHub, and Facebook. Under the icons is a registration form with three input fields: 'username', 'email', and 'password'. Each field has a red 'x' icon on its right side, indicating a validation error. Below the 'password' field is a checkbox labeled 'I'm a teacher' and a link 'or log in'. At the bottom of the form is a blue 'Sign up' button.

Autorizzate uno dei social network o compilate il form di iscrizione

Utilizzo di Repl (<http://repl.it>)



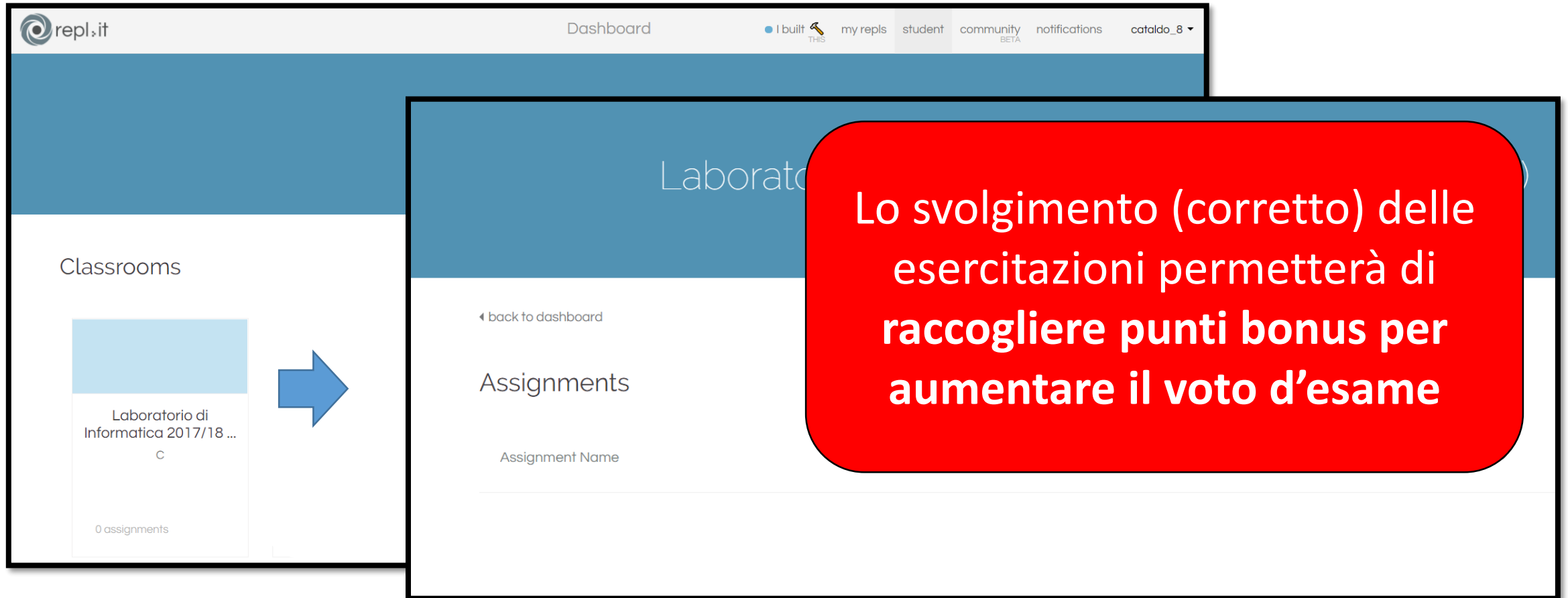
Link di accesso: <https://tinyurl.com/join-repl>

Utilizzo di Repl (<http://repl.it>)

The image shows a screenshot of the Repl.it website. The top navigation bar includes the Repl.it logo, a 'Dashboard' link, and user links for 'I built THIS', 'my repls', 'student', 'community BETA', 'notifications', and a user profile 'cataldo_8'. On the left, under the 'Classrooms' heading, there is a card for 'Laboratorio di Informatica 2017/18 ... C' with '0 assignments'. A large blue arrow points from this card to a larger, detailed view of the same course on the right. This detailed view has a blue header with the course title 'Laboratorio di Informatica 2017/18 - Track B (ITPS) C'. Below the header, there is a 'back to dashboard' link and an 'Assignments' section. The assignments section features a table with columns: 'Assignment Name', 'Date Due', 'Last Activity', and 'All Submissions' (with a dropdown arrow). The table is currently empty.

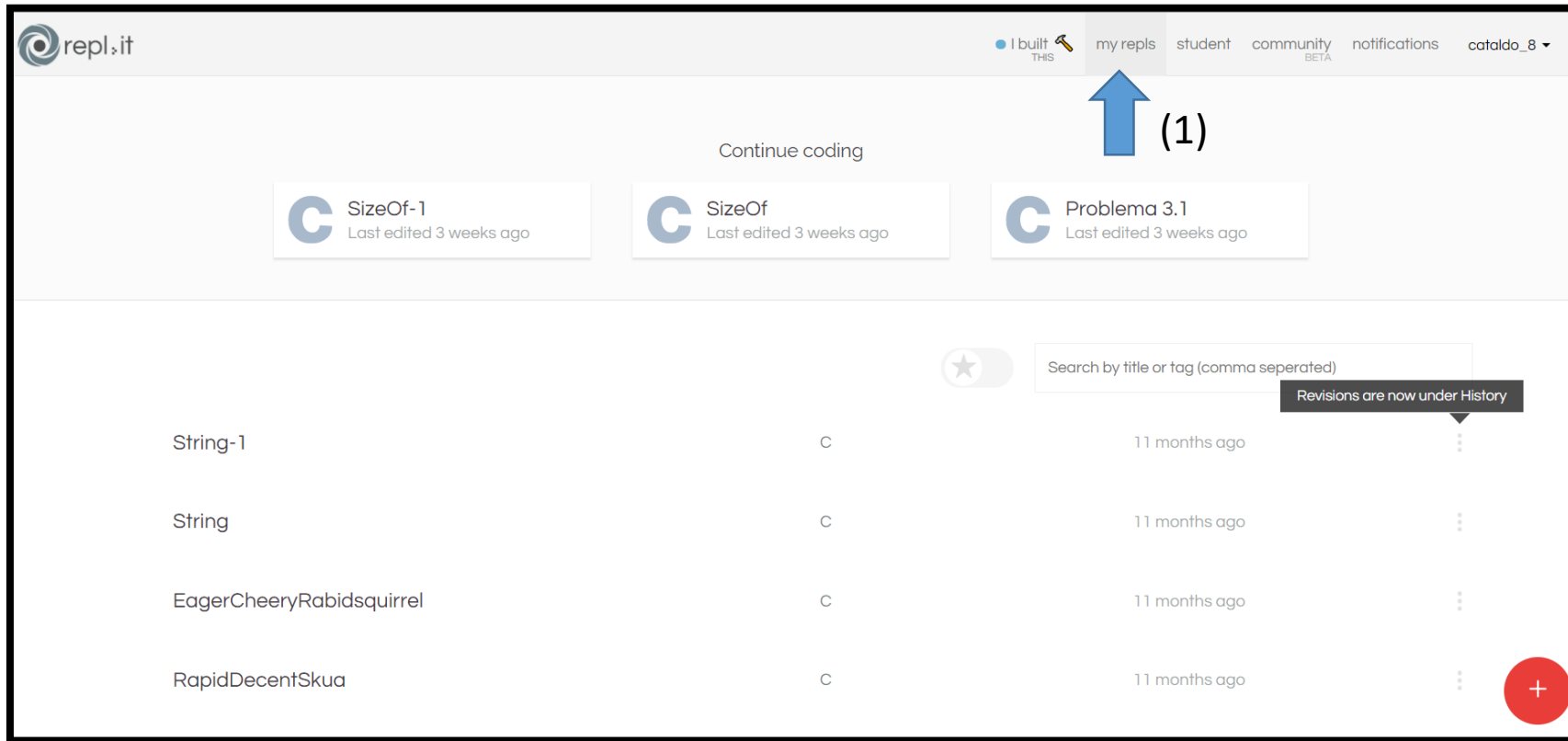
Link di accesso: <https://tinyurl.com/join-repl>

Utilizzo di Repl (<http://repl.it>)



Link di accesso: <https://tinyurl.com/join-repl>

Utilizzo di Repl (<http://repl.it>)



Lo utilizzeremo anche come strumento di esercitazione «libera»
(My Repls → Simbolo '+' in basso → Linguaggio C)

Altre Modalità di Interazione?

facebook



Telegram

Modalità d'esame

- Progetto
- Prova orale



Modalità d'esame

- **Progetto**

- Caso di studio da sviluppare durante il corso (a lezione, durante le ore in Laboratorio, e a casa);
- **Da sviluppare in coppia**

- **Valutazione basata sui temi discussi durante il corso**

- **Correttezza e solidità** del codice
- Buono **stile di programmazione**
- Capacità di **progettare un buon piano di test**
- **Utilizzo di algoritmi di ordinamento** avanzati
- Capacità di fare **astrazione funzionale** e di **progettare correttamente i moduli** che compongono il programma

Modalità d'esame

- **Tre tracce durante l'anno**
 - **Traccia 1: appelli Giugno, Luglio**
 - comunicata a fine Marzo 2018
 - **Traccia 2: appelli Settembre, Novembre**
 - comunicata a fine Luglio 2018
 - **Traccia 3: appelli Febbraio, Aprile**
 - comunicata a Fine Novembre 2018
- **Prova Orale**
 - Discussione del caso di studio e dei contenuti del corso.

Modalità d'esame

```
do{  
    esame();  
}  
while (! (progetto_d_anno()  
    && prova_orale()))
```



Modalità d'esame

```
do{  
    esame ();  
}  
while (! (progetto_d_anno ()  
    && prova_orale ()))
```

Probabile prova parziale a fine corso, che
esoneri dalla parte teorica.
(resta la discussione del caso di studio!)



Domande?



Laboratorio di Informatica
docente: Cataldo Musto
cataldo.musto@uniba.it

Riassumendo

Materiale corso:

<https://elearning.di.uniba.it/course/view.php?id=131>

Piattaforma esercitazioni:

<http://www.repl.it>

Comunicazioni:

via Ada o su cataldo.musto@uniba.it

Esame: progetto+orale (su teoria e caso di studio), eventuali prove parziali per esonerare la parte di teoria