

# Esercitazioni di Laboratorio

## 1. Laboratorio 1

### Esercizio 0

Presentazioni dei Tutor Junior:

- 1) [Il tutor junior](#)
- 2) [JDK \(Java Development Kit\)](#)
- 3) [Linux](#)

### Esercizio 1

**Obiettivo: familiarizzare con l'informazione disponibile online.**

Avviate il browser Firefox tramite interfaccia grafica e mouse oppure da terminale scrivendo il comando **firefox**. Visitare i seguenti siti di interesse didattico per questo corso:

- Visitate il sito dell'aula Taliercio [www.adt.unipd.it](http://www.adt.unipd.it) : dal menu' di sinistra selezionate "**Guide online**". Dall'elenco della documentazione disponibile individuate "Java openjdk 11.0.12". **Questa documentazione sara' disponibile anche durante l'esame, quindi tenete bene presente come accedervi!**
- Visitate il sito moodle del corso Fondamenti di Informatica (A) accessibile da <http://stem.elearning.unipd.it/> . Familiarizzate con le varie sezioni. In particolare consultate nella sezione "Informazioni sul corso" i documenti relativi a "[Calendario delle lezioni e programma svolto](#)" e "[Modalita' d'esame \(LEGGERE ATTENTAMENTE\)](#)".

### Esercizio 2

**Obiettivo: Individuare e configurare l'editor**

Configurare l'editor prescelto (in aula Taliercio e' disponibile gedit) possibilmente con le impostazioni seguenti (tipicamente da menu Preferenze):

- mostrare il numero di riga
  - serve per facilitare la correzione degli errori segnalati dal compilatore
- evidenziare la riga corrente
  - aiuta nella navigazione all'interno del codice
- evidenzia le parentesi corrispondenti
  - aiuta nel verificare la presenza e l'accoppiamento delle parentesi
- ampiezza tabulazione pari a 3 spazi
  - rende il codice ordinato
- inserisci spazi anziché tabulazioni
  - rende più facile l'interscambio di codice sorgente Java

### Esercizio 3

**Obiettivo: Lavorare da riga di comando in Linux (bash shell)**

Aprire un terminale e lavorare da riga di comando, eseguendo i comandi riportati nel file [shell linux](#) .

### Esercizio 4

Usando uno degli editor di testo disponibili sul sistema, scrivere il file **Hello.java**, copiando con precisione quanto segue:

```

1 public class Hello{
2
3     public static void main(String[] args)
4     {
5         // Visualizza un messaggio in output
6         System.out.println("Hello, World!");
7     }
8
9 }

```

Terminata la copiatura, salvare il file e uscire dall'editor. Aprire una shell ed eseguire il comando (di compilazione Java):

### javac Hello.java

Se il comando di compilazione non va a buon fine e vengono segnalati errori, usare di nuovo l'editor per correggere il file Hello.java.

Se la compilazione va a buon fine (ovvero non viene segnalato alcun errore), verificare che sia stato creato il file Hello.class. Aprire una seconda shell ed eseguire il comando (di esecuzione Java):

### java Hello

Verificare che sulla finestra della shell, subito dopo la riga in cui è avvenuta l'esecuzione del comando, venga visualizzato il messaggio:  
**Hello, World!**

Usare nuovamente l'editor per modificare il messaggio da visualizzare in modo che venga appaia un messaggio di saluto a voi stessi. Salvate, compilate ed eseguite. Nel mio caso verrebbe visualizzato il messaggio:

**Hello, Cinzia!**

## Esercizio 5

Utilizzare da terminale il comando **ls** per verificare che i file Hello.java e Hello.class siano presenti. Eliminare il file Hello.class e provare ad eseguire di nuovo il comando di esecuzione Java:

### java Hello

verificando di ottenere una segnalazione d'errore.

Generare nuovamente il file Hello.class mediante l'esecuzione del compilatore Java:

### javac Hello.java

quindi tentare di nuovo l'esecuzione (con **java Hello**), verificando che venga visualizzato il messaggio di saluto con il vostro nome, nel mio caso:

**Hello, Cinzia!**

## Esercizio 6

Scaricare e compilare i file .java nella cartella "Sorgenti con Errori" nella sezione "Attività laboratoriale" (o direttamente a questo [link](#)). Trovare e correggere l'errore. Compilare e verificare che la compilazione non generi errori.

## Esercizio 7

Scrivere un programma che disegni delle facce stilizzate usando i caratteri della tastiera, ad esempio come queste (e magari un po' migliori!). NB: evitate il carattere '\ ' (back-slash) perché darebbe problemi. Vedremo prossimamente perché. Evitate anche il carattere doppio-apice '"', che verrebbe interpretato come fine stringa. Vedremo prossimamente come fare per usarlo nelle stringhe.

```

///|||///  ###|||###  %%%%%
(| O O |) (| O O |) ((~~~~~))
( V )      | 8 |      ))^ ^((
( <_> )    ( VwwwV )   (( .. ))
<____>    <____>      ))--((

```

## Esercizio 8

**Obiettivo: esercitarsi nella definizione e utilizzo di variabili e semplici operatori numerici. Stampa di risultati.**

Scrivere un programma TestDivision che definisca due numeri interi m e n e invii a standard output il risultato e il resto della divisione intera m/n. Compilare e eseguire. Che cosa succede se il divisore vale zero? Scrivere poi un programma che definisca due numeri reali x e y e invii a standard output il risultato della divisione x/y. Compilare ed eseguire. Esegue correttamente? Provare con le coppie di numeri:

x = 7.0 y = 0.0

x = -7.0 y = 0.0

x = 0.0 y = 0.0

## Esercizio 9

**Obiettivo: esercitarsi nella definizione e utilizzo di variabili e semplici operatori numerici. Stampa di risultati.**

Scrivere un programma ProvaVariabili che:

- Definisca una variabile intera larghezza e la inizializzi a 20;
- Definisca una variabile intera lunghezza e la inizializzi a 30;
- Modifichi il valore di larghezza attribuendogli il valore 40;
- Modifichi il valore di lunghezza aggiungendo 15 al suo attuale valore
- Stampi in output il messaggio "L'area del tavolo e' "
- Stampi in output il prodotto dei valori delle variabili larghezza e lunghezza

## Esercizio 10

**Obiettivo: per coloro a cui "Hello world!" esce dagli occhi, o per chi comunque ama le sfide**

Scrivere un programma per scambiare il contenuto di due variabili, ma...

scambiare i valori di a e b **senza usare una variabile temporanea** per memorizzare uno dei due valori (NB: ci sono almeno due soluzioni; una di queste funziona perché in Java le espressioni a destra del simbolo di assegnazione vengono valutate da sinistra verso destra. In c++ tale soluzione non è detto che funzioni perché il c++ analizza le espressioni in un ordine tale da ottimizzare le prestazioni, quindi non necessariamente da sinistra verso destra).

NB: potete usare **solo** assegnazioni e le operazioni che abbiamo visto finora (**somma/sottrazione/moltiplicazione/divisione/modulo**, non potete utilizzare operazioni bit-a-bit, operatori booleani, o altro che non sia stato spiegato a lezione), quindi **tutti possono risolvere il quesito, ma non è banale**.

## QUESTIONARIO CONOSCENZE PREGRESSE (40 min)

<https://stem.elearning.unipd.it/mod/quiz/view.php?id=540437>