

# Programma del Modulo di Fondamenti di Informatica (9CFU)

## Prof. Marco Cococcioni

1. **Sab 30/09/2017 08:30-11:30 (3:0 h)** lezione: Rappresentazione dei numeri naturali. Rappresentazione posizionale e "formula della sommatoria". Algoritmo "MOD&DIV", per la conversione di un numero da base 10 a base beta qualsiasi. Somma di numeri naturali, basata su rappresentazione posizionale. Conversione di numeri da base beta1 a base beta2 qualsiasi, sfruttando la rappresentazione intermedia in base 10. Conversione di numeri da una base potenza di 2 ad un'altra base potenza di due, mediante passaggio intermedio per la rappresentazione in base 2. Prime informazione sull'ambiente per la programmazione in C++ che verrà utilizzato nel corso: CodeBlocks (<http://www.codeblocks.org>)
2. **Sab 07/10/2017 08:30-11:30 (3:0 h)** lezione: Rappresentazione dei numeri interi in base due. Rappresentazione modulo e segno, in complemento alla base e con bias. Rappresentazione dei numeri reali in base due con virgola fissa. Algoritmo parte intera/parte frazionaria.
3. **Sab 14/10/2017 08:30-10:30 (2:0 h)** lezione: Rappresentazione dei numeri reali in virgola mobile, secondo la linea guida IEEE 758 del 1985. Tipo float e tipo double nello C++. Tipo unsigned int e tipo int per i naturali e per gli interi. Il tipo char, per la memorizzazione dei caratteri secondo la tabella ASCII. Come viene vista la memoria RAM dallo C++ (sequenza di celle consecutive da 8 bit). Concetto di variabile in C++ (identificatore dotato di tipo). Concetto di tipo in C++ (insieme di valori ammissibili, dimensione (numero di celle occupate in memoria) e insieme di operazioni che vi possono essere svolte).
4. **Sab 14/10/2017 10:30-11:30 (1:0 h)** esercitazione: Esercizi sulla rappresentazione dei reali in virgola mobile (come passare da un numero reale  $r$  alla sua rappresentazione  $R=\{S,E,F\}$  e viceversa).
5. **Sab 21/10/2017 08:30-11:30 (3:0 h)** lezione: Operatori algebrici in C++ (+, -, \*, /) tra naturali, interi, float e double. Caratteristiche di un operatore: arietà, associatività, priorità. Operatori unari (+, -, ++ e -- prefissi e postfissi). Concetto di costante letterale. Operatore di assegnamento. Valore destro e valore sinistro. Il tipo booleano (bool). Operazioni logiche definite sul tipo booleano. Ingresso da tastiera mediante cin.
6. **Mer 25/10/2017 13:30-15:30 (2:0 h)** lezione: Operatori di confronto (>, <, >=, <=, ==, !=). Istruzione if. Istruzione if-else. Uso di istruzioni if o if-else annidate. Istruzioni iterative: istruzione while e istruzione do-while. Istruzione break.
7. **Mer 25/10/2017 15:30-16:30 (1:0 h)** esercitazione: Esercizi sull'istruzione if-else e sul ciclo while e do-while.
8. **Gio 26/10/2017 13:30-15:30 (2:0 h)** esercitazione: Esercitazione sulle istruzioni if/if-else e sul ciclo while/do-while. Programma per il calcolo della somma dei soli interi positivi fra gli  $N$  inseriti da tastiera. Calcolo della media dei valori positivi. Test per evitare la divisione per zero. Programma per il calcolo del fattoriale di un numero. Per casa: scrivere un programmino per il calcolo della potenza intera non negative di un numero non negative.
9. **Sab 28/10/2017 08:30-11:30 (3:0 h)** lezione: Classe di memorizzazione delle variabili (statica, automatica, dinamica). Come il programmatore vede la memoria logica a sua disposizione (parte dati statica, parte codice, heap e stack). Le funzioni. Passaggio di argomenti per valore, attraverso lo stack. Funzioni che restituiscono un risultato per valore.
10. **Mar 31/10/2017 13:30-15:30 (2:0 h)** lezione: Il tipo derivato riferimento. Riferimento costante. Riferimento costante a costante. Passaggio di argomenti a funzione per riferimento.
11. **Mer 01/11/2017 13:30-16:30 (3:0 h)** lezione: (lezione non tenuta causa festività 1 Novembre)
12. **Gio 02/11/2017 13:30-15:30 (2:0 h)** esercitazione: Esercizi sulle funzioni e sull'uso dei cicli e dei cicli annidati per la stampa a video di forme geometriche. Esercizio sulla soluzione di una equazione del secondo ordine.
13. **Sab 04/11/2017 08:30-11:30 (3:0 h)** lezione: Conversioni esplicite tra tipi fondamentali. Conversioni implicite. Possibilità di esprimere le operazioni mediante notazioni diverse: notazione prefissa, infissa e postfissa. Come vengono tradotte le operazioni in C++, sfruttando la notazione prefissa (esempio  $c = a + b$  viene tradotto come  $c = \text{operator+}(a,b)$ ). Sovrapposizione (overloading) delle funzioni. Esempio: sovrapposizione della funzione per il calcolo di  $b$  elevato ad  $e$  fra naturali e di  $x$  elevato ad  $y$  fra reali. Potenziali ambiguità

introdotte dalla sovrapposizione combinato con il meccanismo delle conversioni implicite. I puntatori in C++. Operatore indirizzo (&) ed operatore di dereferenziazione (\*).

14. [Mar 07/11/2017 15:30-17:30 \(2:0 h\)](#) lezione: Funzione che scambia il contenuto di due variabili, implementata i) usando i riferimenti e ii) i puntatori. Costanti letterali che possono essere assegnate ad un puntatore (0, null e NULL, tutte equivalenti). Operatore = fra puntatori. Operazioni di confronto fra puntatori (operator==, operator!=, operator<).
15. [Mer 08/11/2017 13:30-16:30 \(3:0 h\)](#) lezione: Aritmetica dei puntatori. Funzioni che hanno vettori come argomento. Le funzioni lavorano sui vettori mediante una variabile puntatore contenente l'indirizzo della prima locazione. Necessita' di passare la lunghezza come argomento alla funzione. Conversione della notazione vett[i] in \*(vett+i). Funzione che scambia il contenuto della componente i con quello della componente j all'interno di un vettore. Algoritmo per la ricerca della posizione dell'elemento a valor minimo in un vettore di interi. Ordinamento di un vettore mediante algoritmo del selection sort.
16. [Gio 09/11/2017 13:30-15:30 \(2:0 h\)](#) esercitazione: Esercizi sui vettori. Funzione che stabilisce se la somma degli elementi di un vettore è pari o meno. Funzione che conta quanti elementi pari ci sono in un vettore. Funzione che, dati tre vettori di booleani della stessa lunghezza, calcola lo XOR logico fra ciascun elemento del primo e del secondo vettore e memorizza il risultato nel terzo. Funzione per la ricerca (lineare) di un elemento all'interno di un vettore, se presente.
17. [Mar 14/11/2017 15:30-16:30 \(1:0 h\)](#) esercitazione: Esercizi sulle C-stringhe: re-implementazione della funzione di libreria per la copia di due stringhe (strcpy) e di quella per il confronto di due stringhe (strcmp).
18. [Mar 14/11/2017 16:30-17:30 \(1:0 h\)](#) lezione: Le matrici in C++. Definizione. Come viene allocato in memoria un array bidimensionale. Ogni matrice può essere vista come un vettore, composto dalla concatenazione delle varie righe della matrice. Funzioni che lavorano su matrici. Funzione per la stampa di una matrice. Come passare una matrice ad una funzione.
19. [Mer 15/11/2017 13:30-14:30 \(1:0 h\)](#) esercitazione: Esercizi sulle matrici: funzioni per il calcolo della somma degli elementi sulla diagonale principale, sulla diagonale secondaria.
20. [Mer 15/11/2017 14:30-16:30 \(2:0 h\)](#) lezione: Allocazione di variabili scalari e vettoriali in memoria dinamica. Operatore new e l'operatore delete. Il tipo derivato struct dello C++, per la creazione di contenitori di tipi fondamentali. Vettori di strutture. Funzioni che operano su vettori di strutture. Operatore selettore di membro. Operatore di assegnamento predefinito fra istanze di strutture aventi lo stesso nome.
21. [Gio 16/11/2017 13:30-15:30 \(2:0 h\)](#) lezione: Le liste in C++. Creazione di una lista manualmente. Funzione per la stampa degli elementi di una lista. Funzione per l'inserimento in testa ad una lista.
22. [Sab 18/11/2017 08:30-11:30 \(3:0 h\)](#) lezione: Altre operazioni sulle liste: inserimento in fondo in una lista non ordinata. Estrazione dalla testa, estrazione dal fondo, sempre in liste non ordinate. Inserimento in un lista ordinata, mantenendola ordinata.
23. [Mar 21/11/2017 15:30-17:30 \(2:0 h\)](#) lezione: Regole di collegamento. Identificatori con collegamento interno ed esterno. Programmazione a moduli. La direttiva #include. Unità di compilazione. Come condividere una variabile fra più moduli (parola chiave extern). Come impedire la condivisione (parola chiave static). Lo spazio dei nomi (prima parte).
24. [Mer 22/11/2017 13:30-16:30 \(3:0 h\)](#) esercitazione: Esercizi sui vettori, sulle matrici e sulle stringhe (conta\_cccorrenze).
25. [Gio 23/11/2017 13:30-15:30 \(2:0 h\)](#) lezione: Il tipo pila, implementato come struttura dati pubblica + funzioni globali. Funzioni inip, empty, full, push, pop. Implementazione in due moduli distinti: pila.h (interfaccia) e pila.cpp (implementazione)
26. [Sab 25/11/2017 08:30-09:30 \(1:0 h\)](#) lezione: Visibilità e ancora sullo spazio dei nomi (seconda parte).
27. [Sab 25/11/2017 09:30-11:30 \(2:0 h\)](#) esercitazione: Esercitazione sulle liste (playlist di album musicali).
28. [Mar 05/12/2017 15:30-17:30 \(2:0 h\)](#) lezione: Il costrutto 'class'. Parte pubblica/parte privata. Principio dell'information hiding. Funzioni membro. Funzioni di accesso.
29. [Mer 06/12/2017 13:30-16:30 \(3:0 h\)](#) lezione: Invarianza della rappresentazione interna (classe complesso, in cui si usano modulo e argomento invece che parte reale e parte immaginaria). Funzioni globali che utilizzano istanze di classi definite dall'utente (esempio: funzione globale somma, che somma due complessi e funzione scala, che scala un complesso). Puntatore this. Chiamate concatenate. I costruttori.

30. [Gio 07/12/2017 13:30-15:30 \(2:0 h\)](#) esercitazione: Esercitazione sulla programmazione a moduli mediante struttura dati e funzione globale. La struttura Iscrizione, per la gestione dell'iscrizione dei concorrenti ad una gara di atletica. Conversione della soluzione precedente in una basata su classi (classe Iscrizione).
31. [Sab 09/12/2017 10:30-13:30 \(3:0 h\)](#) lezione: Distruttore. Quando va ridefinito il distruttore. Quando viene chiamato il distruttore (tempo di vita delle variabili). Il costruttore di copia (quando viene invocato, quando va ridefinito, come va ridefinito).
32. [Mar 12/12/2017 15:30-17:30 \(2:0 h\)](#) lezione: Funzioni globali dichiarate amiche della classe, mediante parola chiave friend. Overloading degli operatori, come funzioni membro oppure come funzioni globali. Overloading dell'operatore di assegnamento, esclusivamente come funzione globale.
33. [Mer 13/12/2017 13:30-16:30 \(3:0 h\)](#) lezione: Membri dato costanti nelle classi. Loro inizializzazione mediante l'uso della lista di inizializzazione. Oggetti riferimento membro dato di classi. Classi che usano altre classi e loro inizializzazione mediante lista di inizializzazione. Array di oggetti classe, sullo stack e sullo heap. Funzioni friend. Funzioni membro con attributo const (sono le uniche invocabili su oggetti classe costanti). Membri dato statici e funzioni membro statiche, con applicazione al conteggio delle istanze create e non ancora distrutte di una classe.
34. [Gio 14/12/2017 13:30-15:30 \(2:0 h\)](#) esercitazione: Simulazione di prova pratica: implementazione della classe Grafo (matrice quadrata dinamica). Costruttore, distruttore e overloading di operatori.
35. [Ven 15/12/2017 08:30-11:30 \(3:0 h\)](#) lezione: Operatori bit a bit di or (|), di and (&), di complemento (~), di shift a sinistra (<<) e di shift a destra (>>). La classe ostream e l'istanza cout. Overloading globale dell'operatore di shift a sinistra per stampare a video oggetti definiti dall'utente. Esempio: cout<
36. [Lun 18/12/2017 09:30-12:30 \(3:0 h\)](#) lezione: Operatore ternario. Utilizzo della parola chiave static all'interno di funzioni globali, per creare variabili con contenuto 'persistente', fra una chiamata e l'altra. Suo possibile utilizzo per contare il numero di volte che la funzione e' stata chiamata. Il tipo enumerato. L'istruzione switch. Concetto di funzione che chiama se stessa (programmazione ricorsiva). Implementazione ricorsiva della funzione che calcola il fattoriale e l'n-esimo termine della successione di Fibonacci. Vantaggi e svantaggi dell'implementazione ricorsiva rispetto a quella iterativa.
37. [Mar 19/12/2017 09:30-12:30 \(3:0 h\)](#) esercitazione: Simulazione di prova scritta. La classe Anfiteatro (classe che utilizza al suo interno una lista di interi). Ricevimento collettivo.
38. [Mer 20/12/2017 09:30-11:30 \(2:0 h\)](#) esercitazione: Altra simulazione di prova scritta. Classe Palazzo (array di array).
39. [Mer 20/12/2017 11:30-12:30 \(1:0 h\)](#) lezione: Funzioni inline. Il tipo coda (array circolare a dimensione fissa, gestito con politica FIFO). Applicazione della struttura dati coda alla soluzione della prova d'esame TabTreni.