

Domande Cococcioni

-Domande in base al proprio elaborato

-Mi parli della lista di inizializzazione nelle classi

- cos'è

- quando diventa necessaria

- Esempio di codice

-Definizione e assegnamento di una variabile const (con esempio)

-Oltre che per le costanti quando si utilizza la lista di inizializzazione? (Istanza di classe dentro una classe, reference)

-Mi parli della keyword static

(29 → 30)

-Rappresentare il numero -5 in complemento a 2 su 4 bit

-Il seguente frammento di codice cpp è corretto? Se no, correggilo

```
int a = -1; if (0<=a<=5) cout << "A compreso tra 0 e 5";
```

-Cosa è la regola del cortocircuito (o scorciatoia)

-Literals per il tipo classe

(25 → 28)

-Rappresentare il numero -5 con bias su 4 bit

-Perché viene usato, all'interno dell'elaboratore, la rappresentazione CA2

-Riferimento costante

-Mi commenti il seguente frammento di codice cpp (facendo riferimento alla classe Complesso presente nelle slide):

```
const Complesso c1;  
cout << c1.reale();
```

-Commentare i seguenti frammenti di codice cpp:

- 1) $(a-b) = 7;$
- 2) $(a-=b)=7;$

(29 → 30)

-Cosa accade quando si usa il comando delete anzi che il comando delete[]

-Cosa distingue una lista di interi da un vettore di interi

-Collegamento esterno e collegamento interno

(25 → 26)

-Facendo riferimento alla classe complesso presente nelle slide, parlatemi della ridefinizione dell'operatore di sottrazione

-???

- $(c1-c2)=c3;$ compila o no?

-Commentare la seguente riga di codice cpp:

```
while (cin >>i) cout << i << endl;
```

(30 → 29)

-Cosa mi sa dire sui vettori di oggetti classe

-Operatori di Conversione: Cosa sono, come si definiscono e a cosa servono

-Come mai non ridefiniamo l'operatore di Shift a sinistra (<<) come interno?

(29→30)

-Mi parli dell'algoritmo DIV&MOD, una possibile implementazione in cpp del seguente algoritmo (con la stampa del bit più significativo per primo)

-Implementi una lista doppiamente concatenata e implementi la funzione ricorsiva stampaLista (per la stampa al contrario)

-Ridefinire la somma tra due variabili intere, in modo che la funzione ritorni la maggiore delle due

(28→28)

-Leggere una cstring da tastiera e ristampare l'eco

-Mi parli del tipo enumerato

-Mi parli dei puntatori

(23→27)

-Mi parli del puntatore this nelle classi

-Puntatori nell'ambito delle classi

-Mi parli del tipo Riferimento

-Creare l'istanza di un oggetto nello heap

(21→25)

-(nel main) cout << 'c' + 'a'; cosa succede?

-Se invece che l'intero risultato dall'operazione precedente, volessi vedere un carattere ASCII come dovrei fare?

-Parliamo di overloading di funzioni

-Cos'è uno spazio dei nomi

(26→28)

-Rappresentare 0.4 in virgola fissa

-Meccanismo di ordinamento di vettori

-Come posso ordinare una lista

-Creare una funzione che prende la lista disordinata e ne crea una nuova che contiene gli elementi della lista di partenza messi in ordine

(21.5→26)

-Cosa fa la funzione strcpy + esempio

-Aritmetica dei puntatori (non teoria, esercizi/esempi)

-Mi parli dell'aritmetica dei puntatori

-Supponendo di avere due puntatori a double p e q, facendo cout << p+q, cosa stampa? (non compila non si può fare la somma tra due puntatori) invece cout << p+3;

-Quanti tipi diversi di costruttori conosci

(19→21)

-Cosa succede se facciamo int *p = 7, come possiamo risolverlo? Crea un riferimento a p;

- Riferimento a puntatore (costante e non)

- Operatore di assegnamento nelle classi e test aliasing
- Come si può riscrivere l'operatore this in if(this == &c)
- Cosa succede se si mette void come tipo di ritorno dell'operatore di assegnamento? Perché di solito si mette come tipo di ritorno il tipo della classe?

(24→26)

- Cosa si intende per effetto collaterale? Esempi di funzione con effetti collaterali
- Direttive al preprocessore

(23→26)

- Lettura di una cstringa da tastiera e output a video
- int n = 2; if (n=2) cout << "Pari";
- Cosa si intende per distruttore?
- Static

(18→23)

- Rappresentazione numeri reali in virgola mobile
- Regola del cortocircuito
- Tipo enumerato

(18→18)

- Regola del cortocircuito

(24→)

(21.5→26)

- Quando è opportuno mettere l'attributo const alle funzioni membro? Cosa succede se si fanno due funzioni, una const e una no?

- Come fare per condividere una costante tra due file?

(29→30)

- Lista ordinata in ordine non decrescente come si trova il valore minimo? Che complessità ha? Implementare tale funzione in modo che gestisca il caso di lista vuota in modo corretto.

- Implementare una funzione analoga, ma per il valore massimo. Si può ottimizzare per una lista con i numeri alla fine tutti uguali? (es (1, 3, 4, 7, 7, 7, 7))

(23→27)

- Come si crea il riferimento ad una variabile puntatore? Cosa succede se dereferenzio (con asterisco) un puntatore con valore nullptr? Perché?

- Operatori di conversione per le classi

(18.5→24)

----- non scrivere dopo sta linea che si rompe il font

Domande Perazzo

- Rappresenti in CA2 il numero +30 su 6 bit. A seguito, rappresenti il numero -30

- Date queste 4 variabili in cpp, cosa stampa `cout << x;`

```
unsigned int a = 3;
unsigned int b = 1;
unsigned int c = 5;
unsigned int x = (a|c)&b;
```

- Cosa cambia se nel caso di prima, avevo `unsigned int x2 = (a||c) && b;`
(18→22)

- Dati 2 numeri in CA2, A = 10001110 e B = 10011100, fare la somma e commentarne il risultato. A seguito, fare una seconda somma tra A e B' = 11111100

- Quando è che una somma tra 2 numeri interi su n bit non può essere rappresentata su n bit.

- Mi parli del selection sort

(18,5→22) (rifiutato)

- Rappresenti il numero -1,5 in virgola mobile half precision

- Data la funzione `max`, in cpp, sotto definita, essa è corretta? compila? Se modifico la firma cambiando il return value in `const int &`, cambia qualcosa?

```
int& max(int a, int b) {
    if (a>b) return a;
    else return b;
}
```

```
const int& max(int &a, int &b) {
    if (a>b) return a;
    else return b;
}
```

(30→30)

- Dato il numero in virgola mobile (half precision <1, 01110, 1100000000>, quanto vale in base 10?

- Mi parli dell'algoritmo bubble sort
(20→25)

....

- strlen e sizeof di una cstringa

- cin di una cstringa

(→26)

- R=(0|00010|1110000000) in virgola mobile

- Si può rappresentare lo zero? Quale è il più piccolo numero rappresentabile (in valore assoluto)

- Dato

```
Complesso max(Complesso a, Complesso b){  
    if(a>b) return a;  
    else return b;  
}
```

viene invocato il costruttore di copia? Quando? E passando a e b per riferimento? Perché si mette const?

(29→29)

- Selection Sort su (12, -3, 0, 2, 20, -1, 5, 5)

- Dato

```
int* somma(int a, int b) {
```



```

        int ris = a + b;
        return &ris;
    }

```

Compila? È corretta? Perché? Come potrei modificare la funzione per farla funzionare correttamente senza modificare l'interfaccia?

Usando static funziona?

(18→)

- Dato

```

int main() {
    int n1, n2;
    cin >> n1 >> n2;
    if (n1%n2 == 0)
        cout << "n1 e' divisibile per n2 \n";
    else
        cout << "n1 non e' divisibile per n2 \n";
    return 0;
}

```

Funziona per tutti gli input?

- In questo caso?

```

int main() {
    int n1, n2;
    cin >> n1 >> n2;
    if (n1%n2 == 0 && n2 != 0)
        cout << "n1 e' divisibile per n2 \n";
    else
        cout << "n1 non e' divisibile per n2 \n";
    return 0;
}

```

- Differenza tra ricerca completa e ricerca binaria

- Ricorsione

(21→23)

- Cosa stampa `int a=5, b=5; cout << (a=b) << endl;`

- Cosa stampa `cout << (a+=b+=1) << endl;`

- Controllo aliasing

- Rappresentazione in complemento a due di 30 e -3 su 5 bit. Si può? Su 6 bit?
Sommare i due numeri in binario

(27→29)

- Istruzione break e continue

- Dato

```
for(int i=0; i < 7; i++) {  
    if(i == 2) continue;  
    if(i == 4) break;  
    cout << i << '\n';  
}
```

Cosa stampa?

- $r = -0.2$ come si rappresenta in half precision?

(22.5→/)

- Rappresentare 123 e -2 in complemento a due su 8 bit e sommare i numeri in binario. Il risultato è corretto? Perché?

- Fare la sottrazione tra due numeri in binario. Quando il risultato è corretto

- Quando si verifica overflow in una sottrazione in C2?

(29.5→30)

- Metodi static e variabili membro di una classe. Che classe di memorizzazione ha la variabile membro?

- Rappresentare 0.3 in half precision

(27→28)

- Applicare la ricerca binaria a (6, 3, 3, 2, -1, -10) per trovare l'elemento 1

- Selection sort applicato a (6, 3, 3, 2, -1, -10)

- Risultato di

```
cin >> a >> b >> c;
```

```
cout << (a+b+c)/3 << endl;
```

Inserendo come input 1 1 2. Come si corregge per avere il risultato corretto?

- Stesso input con il codice

```
cout << double(a<<2+b<<2+c<<2)/3 << endl;
```

(20→25)

- Dati i numeri a 8 bit A=(1 0 1 1 1 0 0 0) e di B=(1 0 0 0 1 0 1 1) calcolare la somma. Il risultato è corretto? Fare la prova convertendo in base dieci. In generale quando il risultato non è corretto sommando due naturali

- In numeri rappresentati in complemento a due quando il risultato è corretto?

- Dato

```
int a=3, b=10, c=0;
```

```
cout << (a && b) << '\n' << (!b || c) << '\n' << (!c && a) << '\n';
```

Cosa stampa a video?

- Quanto valgono a, b, c, d dopo la seguente porzione di codice?

```
int a=0, b=a++, c=++a, d=a++;
```

(18→20)(rifiutato)

- Calcolare il valore corrispondente a R=(1|01011|1111000100) dato in virgola mobile.
Cosa succede se moltiplico questo numero per 4?

- Cosa stampa la seguente porzione di codice?

```
int v[15];
```

```
cout << sizeof(v);
```

- Invece cosa stampa questa porzione di codice?

```
cout << sizeof(v)/sizeof(v[0]);
```

(28→29)

...

- Cosa succede nel seguente codice?

```
char str[10];
```

```
cin >> str;
```

(→/)

- Output del seguente codice:

```
int v[10];
```

```
cout << sizeof(v);
```

- Questo?

```
void f(int v[10]){
    cout << sizeof(v);
}
```

- Regola del cortocircuito. Come si sfrutta per evitare la divisione per zero, ad esempio, in un if?

- Come si gestiscono i file in c++? Come vengono gestiti gli errori?

(29.5→27)

- Per cosa si usa la keyword friend?

- Per cosa si usa la keyword static?

- Rappresentare 1.4 in virgola mobile

(20→25)

- Data questa rappresentazione in base 2 su 8 bit A=(1 0 1 0 0 0 1 1) dire ad occhio se è pari o dispari. Convertire in base dieci. Se fosse una rappresentazione in complemento a due? Fare la prova riconvertendo il numero negativo da base dieci a base due.

- Sommare il numero A precedente con B=(0 1 1 1 0 1 0 1) in complemento a due su 8 bit

- In quali casi il risultato è corretto?

- Dato `char* str = new char[20];` la riempio con "hello, world!"
`cout << strlen(str);` Cosa stampa? E `cout << sizeof(str);` Se invece fosse
 stato `char str[20];`

(18→/)

- Significati della parola chiave const. Commentare il caso `const int* p;`

- Bubble sort. Quanti scambi si fanno nel caso peggiore? E nel caso peggiore del selection sort?

- Applicare selection sort al seguente vettore (0, 2, 10, -2, 3, 9, 7)

(19→22)(rifiutato)

- Data una variabile globale `int x = 5;` e una funzione `void f(){ int x=7; cout << x; }` cosa viene stampato in output? Cosa dovrei aggiungere per stampare la variabile globale dentro la funzione? Che tempo di vita hanno le due variabili?

- Cos'è la regola del cortocircuito?

- Dato

```
int a=1, b=0, c=3;  
if ((a+b) && (2*b)) {...}
```

compila? Cosa succede?

In questo caso?

```
if((2*b) && (a+c)) {...}
```

(26→28)

- Cercare l'elemento 3 con la ricerca binaria nel seguente array (10, 4, 7, -1, 5, 0). Ordinarlo con selection sort e applicare la ricerca binaria.

- Quanto vale c?

```
int c = 3*-2+4%0+10/3;
```

In questo caso?

```
int c = 3*-2+4%3+10/3;
```

- Cosa restituisce `double(10/3)`? Questo `double x = 10/3;`?

(20→22)

- Cosa stampa la seguente porzione di codice?

```
enum colore { ROSSO, GIALLO=5, VERDE, BLU };  
c = ROSSO;  
cout << c;  
d=VERDE;  
cout << d;
```

Perché si usano gli enumerator invece che gli int?

- Per cosa si usa la parola chiave static?

es:

```
class C{ static int a;};  
void C::func(){a=1;}
```

Si può fare?

In questo caso cosa cambia?

```
static void C::func2(){a=2;}
```

Se fosse stato così cosa cambia?

```
class C{ static int a;int b;};  
static void C::func2(){a=2;b=2;}
```

(24.5→24)

- Data la seguente struttura coda, quanti elementi si possono memorizzare?

Perché? Cosa potrei fare per poterne memorizzare 10?

```
struct Coda{int v[10]; int testa; int coda;};
```

- Commentare la seguente funzione

```
int max(int a, int b){ return a>b?a:b;}
```

- La seguente modifica è corretta? Perché? Come la si può correggere?

```
int& max(int a, int b){ return a>b?a:b;}
```

- Potrei fare questo nel main (con riferimento all'ultima implementazione con la correzione)?

```
int x=1,y=2; max(x,y)=3;
```

(28→29)

- Rappresentare in single precision il numero $r=1,6$

- Data la rappresentazione in single precision (in cui i puntini indicano tutti zeri) come potrebbe sommare il precedente numero in virgola mobile con $S=(0|0111111|011000000\dots)$? Si è persa precisione in questo caso? In generale?
(30→30L)