UNIVERSITA' DEGLI STUDI ROMA TRE
DIPARTIMENTO DI MATEMATICA E FISICA

Laboratorio di
Programmazione e Calcolo
A.A. 2021-22

Test (Esonero Parziale)

Roma, 12 novembre 2021

1) Le variabili A, B e C sono dichiarate `int`. Se voglio che la variabile C assuma il valore 1 se e solo se le due variabili A e B hanno lo stesso valore, devo scrivere:

☐ C=A=B; ■ C=(A= =B); ☐ (C= =A)=B; ☐ C= = A= =B;

2) Nella directory di lavoro di un utente e' presente solo il file `pippo.cpp` (che contiene un semplice programma main e non richiede l'uso di classi implementate in altri files). Dopo aver eseguito (con successo e senza errori) il comando `g++ -c pippo.cpp` , quali file saranno presenti nella directory di lavoro dell'utente:

☐ `pippo.cpp` e `pippo`
☐ `pippo.o` e `pippo`
☐ `pippo.cpp` e `a.out`
■ `pippo.cpp` e `pippo.o`
☐ `pippo.o` e `a.out`

3) Determinare che cosa viene scritto sul video dopo aver eseguito le seguenti righe di programma:

```
.........................
int a = 3 ;
int b = 4 ;

if(a < b) {
    b = a+b;
 } else {
    b = a * b;
}

cout << b << endl;
.........................
```

☐ Sul video viene scritto: .....7.....

4) Determinare (ed indicarlo chiaramente!) l'errore presente nel seguente programma:

```
#include <iostream>
using namespace std ;

int main()  {

  int a[4] = {10,21,32,43,54} ;

cout << endl;

for(int i=0; i<4; i++) {

  cout << " a["<<i<<"]="<<a[i]<<endl;

}

cout << endl ;

return 1;

}
```

☐ L'errore e': ...L'array e' stato dichiarato per 4 elementi ma nizializzato con 5 elementi...

5) Un programma main (`pippo.cpp`) utilizza oggetti della Classe myfun, dichiarata nel file `myfun.h` ed implementato nel file `myfun.cc` e della Classe mydummy, dichiarata nel file `mydummy.h` ed implementato nel file `mydummy.cc` (e non utilizza metodi di altre classi). Indicare il comando o la serie di comandi necessari per ottenere il file eseguibile di nome `pippo`

.....g++ -c myfun.cc..... g++ -c mydummy.cc.....

.....g++ -c pippo.cpp.....

.....g++ myfun.o mydummy.o pippo.o -o pippo.....

6) Che differenza c'è tra le seguenti dichiarazioni di due metodi
`"void nulla();"` e `"int nulla();"`

☐ nessuna

■ un metodo restituisce un numero intero, l'altro no

☐ la prima è corretta la seconda no

7) Quali sono i valori di i e di j alla fine di ciascuno dei seguenti cicli?

```
int i=0;
do { i++;} while(i<5);

int j=0;
while(j<i){ j++;}
```

☐ i=5; j=4; ☐ i=4; j=4; ■ i=5; j=5; ☐ i=4; j=5;

8) Si considerino le seguenti righe di codice

```
int i = 10;
float  v[i] ;
float* pv=&v[0]  ;
```

Tali righe:

☐ sono prive di senso perché *pv* dovrebbe essere un array di puntatori

■ sono la dichiarazione di un vettore e del puntatore al primo elemento del vettore;

☐ non hanno senso finché non si esplicita il valore dell'intero `i`

9) In un programma è definita la classe C1, in cui è definito il metodo M1(). Si considerino le seguenti dichiarazioni

```
C1 esempioC1 ;
C1 * puntC1 = &esempioC1 ;
```

Per richiamare il metodo M1 si può usare la seguente istruzione:

☐ `esempioC1->M1();` ■ `esempioC1.M1();` ☐ `puntC1.M1();`

10) Si considerino le seguenti righe di codice

```
{ int* miointero1 ;}

{ int intero2=10;
  miointero1=&intero2;}

 cout << miointero1 << " " << *miointero1 << endl ;
```

Cosa vi aspettate che succeda?

■ In fase di compilazione ci sono degli errori

☐ Il programma è compilato e viene creato l'eseguibile che restituisce i valori "10 10"

☐ Il programma è compilato e viene creato l'eseguibile che restituisce valori dei puntatori a miointero1 e 10

11) L'espressione "out of scope" indica

☐ che non si è implementato correttamente un metodo

☐ che la memoria allocata non e' adatta allo scopo

■ l'esterno di un settore in cui viene definito qualcosa

12) Si considerino i seguenti metodi di una classe C1. :

```
void C1::metodo1(int n){ cout<<n<<endl;}

int C1::metodo2(int n) { return n;}
```

Come si richiamano i due metodi affinché sullo schermo compaia lo stesso output, pari a "10", supponendo che `pippo` sia un puntatore ad un oggetto della classe C1 gia' dichiarato?

☐ `pippo->metodo1(10); *pippo->metodo2(10);`

■ `pippo->metodo1(10); cout << pippo->metodo2(10)<<endl;`

☐ `cout << pippo->metodo1(10)<<endl; cout<< pippo->metodo2(10)<<endl;`

13) Si consideri la struttura di una Classe:

■ Gli Attributi caratterizzano la Classe ed i Metodi sono le Funzioni che possono essere applicate agli elementi della Classe;

☐ I Metodi caratterizzano la Classe ed gli Attributi sono le Funzioni che posso essere applicate agli elementi della Classe;

☐ In una Classe e' possibile definire solo Attributi o solo Metodi;

14) Se devo eseguire molte volte una stessa operazione e conosco quante volte l'operazione deve essere ripetuta:

☐ Posso utilizzare l'istruzione while

■ Posso utilizzare l'istruzione for

☐ Devo scrivere piu' volte l'istruzione che deve essere ripetuta

15) Considerate una struttura if ... else if ... else costituita da una serie successiva di blocchi istruzione

```
if(...) {
      .....
    }.else if() {
      .....
    }.else if() {
      .....
 ...................
...} else {
 }
```

☐ Possono essere eseguiti uno o piu' blocchi istruzione a seconda di quali condizioni specificate negli if o else if sono soddisfatte

■ Viene eseguito sempre un solo blocco istruzioni.

☐ Non viene eseguito mai alcun blocco istruzioni.

☐ Il risultato e' imprevedibile

16) Si consideri il seguente codice

```
int i = 5  ;
int j = i  ;
int *q = &i ;
cout << i << " " << j << " " << *q << "   ";
i=8;
cout << i << " " << j << " " << *q << endl;
```

Cosa viene stampato?

☐ 5 5 5 8 8 8 ☐ 5 5 5 8 8 5 ■ 5 5 5 8 5 8

17) Nel mio programma main ho istanziato un oggetto della classe Stoviglia (correttamente definita ed implementata nei file Stoviglia.h e Stoviglia.cc). Se voglio modificare un Attributo (dichiarato private) dell'oggetto gia' istanziato:

- [x] Devo usare un metodo di tipo Set
- [ ] Devo usare un metodo di tipo Get
- [ ] Posso modificare direttamente l'Attributo

18) Un oggetto della Classe Stoviglia (correttamente definita ed implementata nei file Stoviglia.h e Stoviglia.cc) e' stato istanziato in maniera dinamica utilizzando l'operatore new:

```
Stoviglia* MioPiatto = new Stoviglia(...) ;
```

- [ ] la distruzione dell'oggetto istanziato in maniera dinamica dipende dalla quantita' totale di memoria utilizzata dal programma
- [x] nel seguito del programma dovra' essere presente un'istruzione delete
- [ ] il compilatore si occupa in amniera automatica di distruggere gli oggetti non piu' utilizzati

19) E' definita la classe Stoviglia e la classe sua erede Piatto. La classe piatto (file Piatto.h) deve essere dichiarata come:

- [x] `class Piatto : public Stoviglia { ..... } ;`
- [ ] `class Piatto :: public Stoviglia() { ..... } ;`
- [ ] `class Piatto :: Stoviglia() { ..... } ;`
- [ ] `class Piatto { ..... } ;`

20) Sempre sulla classe Piatto, erede di Stoviglia. Se Massa è il primo attributo di Piatto e di Stoviglia, come si fa implementare il costruttore di un Piatto e, dunque, di una Stoviglia di stessa massa?

- [x] `Piatto :: Piatto(float mass,.....) : Stoviglia(mass,....) { ..... }`
- [ ] `Piatto :: Piatto(float mass,.....)`
  `{ Stoviglia.Massa=mass; .....}`
- [ ] `Piatto :: Piatto(float mass=Stoviglia.Massa,.....)`
  `{Massa=mass; .....}`