

Problemes 1: Programació en C++

Estructura de Dades - Curs 2020/2021
Grau d'Enginyeria Informàtica
Facultat de Matemàtiques i Informàtica

UNIVERSITAT DE
BARCELONA

Objectiu

Aquests problemes estan enfocats per a que practiqueu i tingueu uns coneixements bàsics del llenguatge C++. Els problemes estan ordenats segons la seva dificultat. Si us plau, feu els problemes en l'ordre suggerit.

1 Problemes bàsics de llenguatge C++

1. Escriu un programa `Fun.cpp`. Creeu un mètode `main()` que retorni l'enter 0 i escrigui per pantalla:

[illegible]

O qualsevol altre missatge divertit.

```
1 #include <iostream>
2
3 int main() {
4     // Write your code here
5     return 0;
6 }
```

2. Escriu un programa amb un `main()` que demani a l'usuari el seu pes en kg i cridi un mètode `toPounds(float kg)` que retorni el pes en pounds.
3. Escriu un programa amb un `main()` que demani a l'usuari un número i indiqui per pantalla si el nombre és parell o senar.

4. Escriu un programa que demani a l'usuari un número entre l'1 i el 7, i la sortida per pantalla serà indicar el dia de la setmana a que correspon aquest número.
5. Escriu un programa que calculi el sou que li correspon a un treballador d'una empresa que cobra 40.000 euros anuals, el programa ha de realitzar els càlculs en funció dels següents criteris:
 - Si porta més de 10 anys en l'empresa se li aplica un augment del 10%.
 - Si porta menys de 10 anys però més que 5 se li aplica un augment del 7%.
 - Si porta menys de 5 anys però més que 3 se li aplica un augment del 5%.
 - Si porta menys de 3 anys se li aplica un augment del 3%.

L'antiguitat és una dada que se li demana a l'usuari per teclat.

6. Escriu un programa que demani a l'usuari un número entre l'1 i el 12, i la sortida per pantalla serà indicar el mes en text a que correspon aquest número.
7. Teniu un germà petit que està aprenent les taules de multiplicar i el voleu ajudar, prepareu un programa que li mostri multiplicacions i li demani per entrar el resultat i depenent si l'ha encertat o no mostreu un missatge felicitant-lo o deixant que ho torni a intentar com a màxim 5 cops i sinó mostrant-li el resultat.
8. Escriu un programa que demani un número de l'1 al 10 i escrigui per pantalla la taula de multiplicar d'aquest número.
9. Escriu un programa que demani la nota que ha tret el vostre germà a l'examen de les taules de multiplicar i mostri *suspès*, *suficient*, *bé*, *notable* o *excel·lent*.
10. Escriu un programa que demani 10 números y els guardi en un array estàtic. Un cop guardats, indiqueu per pantalla quin és el més gran i quin és el més petit en aquest array.
11. Escriu un programa que demani entrar per consola 15 lletres, les guardi en un array estàtic i compti quantes vegades es troba repetida cada lletra.
12. Escriu un programa que donat un array de n posicions. Si la seva mida és parell, invertirà els elements de cada meitat de l'array. Per exemple: $v=[1][2][3][4][5][6]$ $v(\text{invertit})=[3][2][1][6][5][4]$
13. Crea la classe **Pacient** que contingui el **nom**, **edat**, **alçada** i **pes** del pacient. Crea un mètode a la classe **Pacient** per calcular el BMI (Body Mass Index) amb l'alçada i el pes i que escrigui per pantalla si el BMI és molt baix, baix, normal, alt o molt alt. El càlcul és relativament simple: consisteix a dividir el pes, expressat en quilos, entre l'alçada, en metres, elevada al quadrat (kg/m^2). El BMI resultant indica si el pes és correcte en funció d'aquesta alçada, o si és massa baix o elevat.
 - Generalment es considera satisfactori un BMI que se situa entre 18,5 i 24,9.
 - Si el BMI és inferior a 18,5, probablement sigui massa prim.
 - Si és igual o superior a 25, tens uns quilos de més.
 - Si el BMI està per sobre de 30, podem parlar d'obesitat.

Atenció: l'BMI és vàlid només per a majors de 18 anys; no és aplicable per a adolescents si no s'introdueix un factor de correcció.

2 Problemes de classes i objectes en C++

14. Crea un nou projecte NetBeans i definiu una classe `Time` que permeti guardar l'hora, minut i segon. L'hora ha d'estar en el rang de valors de 0 a 23, minut té un rang de valors de 0 a 59 i segon té un rang de 0 a 59. A la implementació de la classe, fareu els següents mètodes:

- Un getter per a cada atribut
- Una funció setter per a cada atribut
- Un constructor per defecte per a la classe
- Un constructor amb paràmetre per a la classe. Aquest constructor recollirà les dades pels tres atributs.
- Una funció `nextSecond` que retornar un objecte `Time` amb un segon més que l'actual.
- Una funció `print` que mostri per pantalla l'hora, minut i segon en el següent format: hh:mm:ss

A més a més, implementeu un `main.cpp` que crei dues instàncies de `Time`. La primera d'elles ha d'estar creada estàticament amb el constructor amb paràmetres, la segona d'elles l'heu de crear dinàmicament. Afegiu dues hores a cada instància i imprimeu per pantalla el seu contingut.

15. Definiu un nou projecte NetBeans i definiu la classe `Usuari`. La classe `Usuari` té dues dades membre: Nom d'Identificació de l'Usuari (conegut com ID) i el Nombre d'Identificació Personal (conegut com a PIN). Per a efectes d'aquest exercici, es considerarà vàlid un ID que tingui una longitud mínima de 6 caràcters; i en el cas del PIN el valor vàlid serà qualsevol nombre major que zero. Implementeu els següents mètodes a la classe:

- Una funció setter per introduir les dues dades membre.
- Una funció setter per a cada dada membre.
- Un constructor per defecte per a la classe.
- Un constructor amb paràmetres per a la classe. Aquest constructor recollirà les dades per tots els atributs.
- Una funció `print` que mostri per pantalla les dades membre de l'objecte.

A més a més, implementeu un `main.cpp` que faci les següents accions:

- Definir i inicialitzar un primer objecte estàticament amb el constructor per defecte
- Definir i inicialitzar un segon objecte estàticament amb el constructor amb paràmetres
- Definir i inicialitzar un tercer objecte dinàmicament amb el constructor per defecte
- Definir i inicialitzar un quart objecte dinàmicament amb el constructor amb paràmetres
- Demanar les dades pel teclat per definir valors als objectes construïts amb el constructor per defecte.
- Guardar els quatre objectes en un array estàtic de 4 posicions
- Imprimir la informació de cada objecte de l'array.

16. Crea un projecte i definiu una classe `Complex` que permeti emmagatzemar la part real i la part imaginària d'un nombre. Els valors per defecte estaran a 0.0 i els atributs de la classe són privats. Es necessari implementar els següents mètodes a la classe:

- Un getter per a cada atribut.
- Una funció setter per a cada atribut.
- Un constructor per defecte per a la classe.
- Un constructor amb paràmetre per a la classe. Aquest constructor demanarà les dades pels dos atributs.
- Una funció `Complex & addInto(Complex & another)` que permeti sumar la informació d'un complex al complex actual (`this`).
- Una funció `Complex addReturnNew(double real, double imag) const` que permeti retornar una nova instància del complex amb la suma. L'actual (`this`) no ha de canviar.
- Una funció `print` que mostri per pantalla les dades del complex (`real`, `imag`).

A més a més, implementeu el `main.cpp` que usi aquesta nova classe `Complex` i que permeti guardar un array estàtic de 10 instàncies de tipus complex. Ompliu l'array d'instàncies i, posteriorment, mostreu el contingut de cada objecte de l'array per pantalla.

17. Crea un projecte que contingui tres classes. La classe `Pacient` definida a l'exercici anterior, una nova classe anomenada `LlistatPacients` que permeti emmagatzemar Pacients en un array dinàmic i tingui dos mètodes: un mètode per anomenat `afegirPacient(Pacient pac)` i un altre `eliminarPacient(String nom)`, i un `main()`. En el `main` es farà un petit menú amb dues opcions:

1. Afegir pacients
2. Eliminar Pacient.

Les dades s'introduiran per teclat.

18. Modifica el problema anterior, permetent que tingui una tercera opció al menú que permeti afegir pacients des d'un fitxer de text. El nom del fitxer es llegeix des de teclat i cada fila del fitxer contindrà la informació d'un pacient. S'han d'introduir tots els pacients del fitxer a `LlistatPacients.cpp`
19. Modifica el problema anterior i incorpora una quarta opció de menú que permeti abocar tota la informació dels pacients en un fitxer de text. El nom del fitxer a abocar es demanarà a l'usuari per teclat. Cada fila del fitxer de sortida tindrà la informació d'un pacient, separant cada camp (atribut) per un tabulador.