

1. Cat ocupa o variabila de tip int (se ia un considerare un compilator din VS2015 pe 32/64 de biti) ?

4 bytes

2.Cat ocupa o variabila de tip short int (se ia un considerare un compilator din VS2015 pe 32/64 de biti) ?

2 bytes

3.Cat ocupa o variabila de tip long int (se ia un considerare un compilator din VS2015 pe 32/64 de biti) ?

4 bytes

4.Cat ocupa o variabila de tip long long int (se ia un considerare un compilator din VS2015 pe 32/64 de biti) ?

4 bytes

5.Cat ocupa o variabila de tip bool (se ia un considerare un compilator din VS2015 pe 32 de biti) ?

1 byte

6.Cat ocupa o variabila de tip float (se ia un considerare un compilator din VS2015 pe 32 de biti) ?

4 bytes

7.Cat ocupa o variabila de tip double (se ia un considerare un compilator din VS2015 pe 32 de biti) ?

8 bytes

8.Cat ocupa o variabila de tip long double (se ia un considerare un compilator din VS2015 pe 32 de biti) ?

8 bytes

9.Cat ocupa o variabila de tip char (se ia un considerare un compilator din VS2015 pe 32 de biti) ?

1 byte

10. Cat ocupa o variabila de tip pointer la int - int *(se ia un considerare un compilator din VS2015 pe 32 de biti)?

4 bytes

11. Cat ocupa o variabila de tip pointer la char- char *(se ia un considerare un compilator din VS2015 pe 32 de biti) ?

4 bytes

12. Ce reprezinta cuvantul cheie this in C++ ?

Este un pointer primit ca primul parametru de toate metodele non-statica din clasa si care contine adresa obiectului care a apelat metoda

13. Ce reprezinta this in constructor ?

Este un pointer ce contine adresa obiectului ce urmeaza sa fie construit

14. Ce reprezinta this in destructor ?

Este un pointer ce contine adresa obiectului ce urmeaza sa fie distrus

15. Ce reprezinta this intr-o functie membra ?

Este un pointer ce contine adresa obiectului care a apelat metoda.

16. Ce reprezinta this intr-o functie statica ?

O functie statica nu este asociata cu niciun obiect, deci nu primeste pointerul this.

17. Ce reprezinta this in constructorul de copiere ?

Este un pointer catre obiectul ce urmeaza sa fie creat.

18. Ce reprezinta this in cadrul supraincarcarii operatorului = ?

Este un pointer catre obiectul ce urmeaza sa fie modificat.

19.Ce este o functie membra ?

Este o functie al carei definitii sau prototip se afla in definitia unei clase ca orice alta variabila si opereaza pe orice obiect/instanta

a acelei clase si are acces la toti membrii clasei.

20.Ce este o functie statica ?

Este o functie membra, independenta de obiectele clasei.O functie statica poate fi apelata chiar daca nu exista nicio instanta a clasei.Ele sunt apelate

folosind numele clasei si operatorul de rezolutie.

21.Ce este un atribut static ?

Un atribut static este un atribut ce nu face parte din obiect, in schimb el este impartit de toate obiectele clasei.Nu poate fi initializat in interiorul unei clase

22.Ce este un atribut constant ?

Un atribut constant este un atribut ce nu poate fi modificat in timpul executiei o data ce a fost initializat.

23.Cand se poate initializa un atribut constant ?

Un atribut constant se poate initializa doar la crearea obiectului, in lista de initializare a constructorului.

24.Ce reprezinta supraincercarea ?

Supraincercarea reprezinta abilitatea de a avea mai multe functii cu acelasi nume dar cu o lista de parametrii diferiti.

25.Care este utilitatea supraincercarii ?

Putem avea mai multe functii cu acelasi nume, care fac acelasi lucru sau lucruri diferite, dar pentru tipuri de date diferite si chiar si tipuri de date create de noi.

26.Ce reprezinta supradefinirea ?

Supradefinirea este abilitatea de a avea 2 functii, una in clasa de baza si una in clasa derivata, cu acelasi nume si aceeasi lista de parametrii dar care sunt

diferite prin implementare.

27.Care este utilitatea supradefinirii ?

Ne permite sa avem 2 functii cu acelasi nume si aceeasi parametrii, una in clasa de baza si una in clasa derivata, care sunt diferite prin implementare.

28.Ce este o functie virtuala ?

O functie virtuala este o functie membra ce astepti sa fie redefinita intr-o clasa derivata.Cand te referi la un obiect al clasei derivate printr-un

pointer la clasa de baza apelezi varianta din clasa derivata a functiei ce a fost declarata virtuala in baza.

29.Prin ce se implementeaza conceptul de Polimorfism in C++ ?

Polimorfismul in C++ se implementeaza prin functii virtuale, supradefinire,supraincarcarea operatorilor si upcasting.

30.Care sunt modificatorii de acces si ce vizibilitate ofera datelor/functiilor membre in cazul derivarii claselor?

In C++ exista 3 modificatori de acces:public, private si protected.

Atributele si metodele declarate publice pot fi accesate din orice parte a programului.

Atributele declarate protected pot fi accesate doar din interiorul clasei si din clasele derivate.

Atributele declarate protected pot fi accesate doar in interiorul clasei.

31.In ce context este util modificatorul de acces private ?

Modificatorul private este util la incapsularea datelor.Acesta nu permite accesul altor clase, nici macar a celor derivate, la atributele clasei.

32.Care sunt tipurile de constructori in cadrul unei clase si ce rol are fiecare?

Exista 3 tipuri de constructor:default,cu parametrii si de copiere.

Constructorul default construiește un obiect initializand atributele cu valori constante, ce nu sunt primite ca parametrii.

Constructorul cu parametrii construiește un obiect cu parametrii primiti de la apelator.

Constructorul de copiere creaza copii a unor obiecte care sunt transmise sau returnate prin valoare sau creaza un obiect nou, initializand attributele cu valorile altui obiect.

33.Care este diferenta intre rolul operatorului = si cel al constructorului de copiere?

Operatorul = se apeleaza pentru 2 obiecte existente deja.

Constructorul de copiere creaza obiectul si atribuie atributelor valorile altui obiect deja existent.

34.Cand este apelat constructorul de copiere ?

Constructorul de copiere se apeleaza atunci cand transmitem sau returnam obiecte prin valoare sau cand vrem sa construim un obiect nou care sa fie o copie a unui alt obiect deja existent.

35.Cand este apelat operatorul = ?

Operatorul = este apelat atunci cand vrem sa atribuim unui obiect deja existent valorile unui obiect deja existent.

36.Ce este un memory leak ?

Un memory leak se produce atunci cand nu dezalocam spatiu pentru un obiect de care nu mai avem nevoie.

37.Ce este un dangling pointer?

Este un pointer ce indica spre o zona de memorie ce a fost eliberata deja.

38.Care este rolul destructorului ?

Rolul destructorului este de a dezaloca memoria alocata la crearea sau in timpul folosirii obiectului.

39.Cand se apeleaza destructorul ?

Destructorul se apeleaza la sfarsitul blocului de cod in care a fost declarat obiectului.

40.Ce este memoria HEAP?

Heap este o portiunea de memorie unde putem aloc si dezaloca spatiu in timpul executiei.

41.Cum se alocă spațiu de memorie în HEAP?

Spațiu în Heap se alocă folosind operatorul new.

42.Cum se eliberează memoria în HEAP ?

Memoria alocată în Heap se eliberează folosind operatorul delete.

43.Cum se generează un memory leak ?

Un memory leak se generează atunci când distrugem un obiect și nu eliberăm memoria din heap alocată pentru el.

44.Care este rolul funcțiilor accesori în cadrul clasei?

Funcțiile accesori au rolul de a oferi o interfață prin care putem controla accesul la membrii privați ai clasei.

45.Ce rol au funcțiile friend în cadrul claselor și care sunt caracteristicile acestora ?

Funcțiile friend oferă acces la membrii privați ai clasei unor funcții sau clase care nu se află în clasă.

Funcțiile friend nu primesc pointerul this.

46.Ce reprezintă conceptul de încapsulare ?

Este conceptul prin care restricționăm accesul direct la membrii unei clase.

47.Care este diferența dintre supradefinire și supraincarcare?

Supradefinirea necesită existența unei ierarhii de clase și cele 2 funcții trebuie să se afle în clase diferite.

Supraincarcarea poate fi folosită pentru 2 funcții din aceeași clasă și chiar și în afara claselor între orice 2 funcții.

48.Cum se poate afla dimensiunea unui vector de char?

Folosind funcția strlen()

49.Care e vizibilitatea implicita in cadrul unei clase?

Private

50.Enumerati trei functionalitati ale operatorului * in C++.

Declararea unui pointer.

Extragerea valorii de la o adresa.

Inmultire.

51.De ce se folosesc in general functiile de acces in detrimentul vizibilitatii publice?

Pentru a nu oferi acces direct la membrii functiei si pentru a controla modul in care acestea sunt modificate.

52.De ce tip (clasa) este obiectul cin?

istream

53.De ce tip (clasa) este obiectul cout?

ostream

54.Ce reprezinta prescurtarea STL?

Standard Template Library

55.Dati exemplu de un container STL.

list, set, map

56.Dati exemplu de un algoritm STL.

swap(interschimba valorile pentru 2 obiecte),for_each, copy

57.Prin ce tipuri de metode se poate supraincarca un operator ?

Un operator se poate supraincarca printr-o functie membra sau o functie in afara clasei

58. Prin ce tipuri de metode se poate supraîncărcare operatorul + pentru (obiect + int) ?

Operatorul + pentru (obiect + int) se poate supraîncărca printr-o funcție membră sau o funcție în afara clasei.

59. Prin ce tipuri de metode se poate supraîncărcare operatorul + pentru (int + obiect) ?

Operatorul + pentru (int + obiect) se poate supraîncărca DOAR printr-o funcție în afara clasei.

60. Când trebuie utilizat "friend" la supraîncărcarea operatorilor ?

Atunci când avem atribute private și nu avem funcții de acces.

61. Ce realizează constructorul de copiere ?

O copie a unui obiect.

62. Cum se apelează destructorul ?

Destructorul se apelează în ordinea inversă construirii obiectelor.

63. Care este diferența dintre o variabilă și un atribut ?

Un atribut este un câmp al unei clase.

64. În ce context este util modificatorul de acces protected ?

Protected este folosit pentru a oferi acces direct unei clase derivate la membri clasei de bază, păstrând atributele clasei de bază private pentru restul programului.

65. Care este diferența dintre specificatorii class și struct ?

Class are ca specificator default de acces private iar struct are public

66. Ce reprezintă o clasă ?

Este o structură de date definită de utilizator care conține date și funcții iar accesul la aceste date se face prin specificatori de acces.

67.Ce reprezinta un obiect ?

Un obiect este o instanta a unei clase si contine toate campurile din clasa din care a fost instantiat.

68.Ce reprezinta o instanta a unei clase ?

Un obiect cu campurile din clasa respectiva

69.Ce este un constructor ?

O functie ce creaza un obiect

70.Ce reprezinta termenul de up-casting ?

Folosirea unui pointer de tipul clasei de baza pentru a gestiona obiecte din clasa derivata

71.Ce reprezinta termenul de down-casting ?

Folosirea unui pointer de tipul clasei derivate pentru a gestiona obiecte din clasa de baza

72.Ce este un framework de clase ?

O ierarhie de clase

73.Pot fi definite metode constante ?

NU

74.Care este ordinea de apel a constructorilor in cadrul ierarhiilor de clase ?

De la constructorul din baza catre cel din clasa derivata

75.Care este ordinea de apel a destructorilor in cadrul ierarhiilor de clase ?

De la cel din clasa derivata catre cel din clasa de baza.

76.Ce rol au functiile virtuale in cadrul ierarhiilor de clase ?

Anunta ca ele vor fi supradefinite in clasele derivate.

77.Ce este o functie virtuala pura ?

Este o functie fara corp ce urmeaza a fi supradefinita intr-o clasa derivata

78.Ce este o clasa abstracta ?

Ce contine minim o functie virtuala pura.

79.Ce restrictii impune o clasa abstracta ?

O clasa abstracta nu poate fi instantiata.(Nu se pot crea obiecte de acel tip)

80.Cum se realizeaza mostenirea multipla in C++ ?

Mentionand clasele din care se va deriva in ordinea in care se vrea sa se apeleze constructorii.

81.Ce sunt functiile inline ?

Functii ale caror apel vor fi inlocuite de codul lor la compilare

82.Ce reprezinta conceptul de is a ? Exemplificati

Derivare Ex:Autovehicul is a Masina

83.Ce reprezinta conceptul de has a ? Exemplificati

Includere Ex:Spital has a Medic

84.Ce este mostenirea virtuala?

De chestia asta nu a zis nimic la curs si nici nu am gasit mare lucru pe net si nu am inteles ce este

85.In ce moment are loc efectiv supradefinirea?

In timpul executiei

86. In ce moment are loc efectiv supraincercarea?

La compilare

87. Care e vizibilitatea implicita in cadrul unei structuri?

Public

88. Cum se poate modela o relatie de tip 1:M intre doua clase?

Habar nu am ce vrea sa zica aici

89. Cum se poate face conversia baza-derivat?

prin `dynamic_cast<>()`

90. Cum rezolvam problema mostenirii in romb (deadly diamond of death)?

Evitam mostenirea multipla

91. In ce ordine se apeleaza constructorii in cazul unei derivari multiple?

In ordinea in care s-a facut derivarea

92. Cum se poate rezolva problema nefunctionarii unei clase sablon pe un anumit tip de data?

Din nou nu inteleg ce vrea sa zica prin intrebarea asta

93. Cum poate deveni un iterator invalid?

Daca adaugam sau stergem elemente din container (nu sunt 100% sigur)

94. Cum se defineste un pointer prin care se poate gestiona orice tip de zona de memorie ?

`char*`

95. Cum se poate accesa zona privata a unui obiect in C++ ?

Printr-o functie accesoriu sau printr-un pointer la `char`

96.Ce reprezinta termenul de late-binding ?

Alegerea functiei ce urmeaza sa fie executata la executie

97.Ce reprezinta termenul de early-binding ?

Alegerea functiei ce urmeaza sa fie executata la compilare.

98.Care este utilitatea functiilor virtuale ?

Putem crea vectori cu obiecte de tip diferit,daca ele fac parte din aceeaasi ierarhie, si prin intermediul functiilor virtuale putem decide

ce functie apelam pentru fiecare obiect din vector.

99.La ce pot fi folosite clasele abstracte ?

La derivare

100.Poate fi folosit un pointer la o clasa abstracta pentru a gestiona obiecte de tipul claselor derivate ?

Da

101.Care este utilitatea atributelor constante ?

Putem defini atribute ce nu vor fi modificate dupa initializare

102.Cum se defineste o metoda care sa nu primeasca pointerul this ?

Statica