Clase – Supraîncărcare Operatori

[Observații: 1](#_Toc466837164)

[Operator= 1](#_Toc466837165)

# Observații:

* Documentul este gândit pentru studenții care au participat la seminar. Este posibil sa întâmpinați dificultăți in parcurgerea / înțelegerea lui daca nu ați fost prezenți.
* Eventualele dezacorduri sau greșeli de scriere au fost făcute intenționat, cu rolul de a vă binedispune în timpul citirii acestui document. ☺

# Operatori unari

1. **! (**[**logical NOT**](https://msdn.microsoft.com/en-us/library/1k6w8551.aspx)**)**
2. & ([address-of](https://msdn.microsoft.com/en-us/library/64sa8b1e.aspx))
3. ~ ([one's complement](https://msdn.microsoft.com/en-us/library/dxt4z71k.aspx))
4. \* ([pointer dereference](https://msdn.microsoft.com/en-us/library/fw63e3c3.aspx))
5. + ([unary plus](https://msdn.microsoft.com/en-us/library/ktz6ya2s.aspx))
6. - ([unary negation](https://msdn.microsoft.com/en-us/library/ktz6ya2s.aspx))
7. **++ (**[**increment**](https://msdn.microsoft.com/en-us/library/dy3d35h8.aspx)**)**
8. **-- (**[**decrement**](https://msdn.microsoft.com/en-us/library/dy3d35h8.aspx)**)**
9. **pperator cast**

|  |  |
| --- | --- |
|  | Further reading: <https://msdn.microsoft.com/en-us/library/f672kxz8.aspx> |

# Operatori binari

1. , Comma
2. != Inequality
3. % Modulus
4. %= Modulus/assignment
5. & Bitwise AND
6. && Logical AND
7. &= Bitwise AND/assignment
8. \* Multiplication
9. \*= Multiplication/assignment
10. + Addition
11. += Addition/assignment
12. - Subtraction
13. –= Subtraction/assignment
14. -> Member selection
15. –>\* Pointer-to-member selection
16. / Division
17. /= Division/assignment
18. < Less than
19. << Left shift
20. <<= Left shift/assignment
21. <= Less than or equal to
22. = Assignment
23. == Equality
24. > Greater than
25. >= Greater than or equal to
26. >> Right shift
27. >>= Right shift/assignment
28. ^ Exclusive OR
29. ^= Exclusive OR/assignment
30. | Bitwise inclusive OR

|  |  |
| --- | --- |
|  | Further reading: <https://msdn.microsoft.com/en-us/library/czs2584d.aspx> |

# Assignment Operator (Operator=)

Further reading:

* http://en.cppreference.com/w/cpp/language/operators

1. Modificați codul din metoda main ca mai jos.

|  |
| --- |
| **void** main**()**  **{**  **int** marks**[]** **=** **{** **10,9** **};**  Student s**(**"Nume"**,** **21,** **2,** marks**);**  *//Constructor de copiere*  *//OMetodaSimpla(s); //apeleaza constructorul de copiere*  *//Student s2 = s; //apeleaza constructorul de copiere*  Student s2**(**"Nume2"**,** **22,** **2,** marks**);**  *//Operator=*  s2 **=** s**;** *//apeleaza operator=*  *//s2.operator=(s); //echivalent cu s2 = s;*  **}** |

1. Rulați programul și observați eroarea afișată. Încercați să determinați linia de cod la care se produce eroarea.
2. Adăugați în clasa Student implementarea pentru operatorul egal

|  |
| --- |
| Student **&** **operator=(const** Student **&** source**)**  **{**  cout **<<** "Operator =" **<<** endl**;**  *//1.*  **this->**varsta **=** source**.**varsta**;**  **this->**nrNote **=** source**.**nrNote**;**  *//2. copiere atribute \**    *//Nume*  *//a) stergere spatiu existent*  **if** **(this->**nume **!=** NULL**)**  **delete[]** **this->**nume**;**  **if** **(**source**.**nume **==** NULL**)**  **this->**nume **=** NULL**;**  **else**  **{**  *//b) alocare spatiu*  **this->**nume **=** **new** **char[**strlen**(**source**.**nume**)+1];**  *//c) copiere*  strcpy**(this->**nume**,** source**.**nume**);**  **}**  *//Note*  *//a) stergere spatiu existent*  **if** **(this->**note **!=** NULL**)**  **delete[]** **this->**note**;**  *//b) alocare spatiu*  **this->**note **=** **new** **int[this->**nrNote**];**  *//c) copiere*  **for** **(int** i **=** **0;** i **<** **this->**nrNote**;** i**++)**  **this->**note**[**i**]** **=** source**.**note**[**i**];**  **}** |