

Functii const

Pentru a creea un obiect constant se foloseste cuvantul cheie const inainte. Daca se incearca modificarea atributelor unui obiect const atunci primim eroare de compilare.

```
const Nume_Clasa Nume_Obiect;
```

- O functie declarata constanta poate fi apelata atat pe obiecte constante cat si pe obiecte non-const.
- Daca un obiect este declarat const trebuie initializat la momentul declararii.

Functiile const sunt folosite pentru a nu permite modificarea obiectului. Este o metoda folositoare pentru a nu modifica accidental membrii unui obiect.

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

class Test {
    int value;
public:
    Test(int v = 0) {value = v;}

    // We get compiler error if we add a line like "value = 100;"
    // in this function.
    int getValue() const {return value;}
};

int main() {
    Test t(20);
    cout<<t.getValue();
    return 0;
}
```

Output:

20

Exemplu de program cu erori:

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

class Test {
    int value;
public:
    Test(int v = 0) {value = v;}
    int getValue() {return value;}
};

int main() {
    const Test t;
    cout << t.getValue();
    return 0;
}
```

Output:

passing 'const Test' as 'this' argument of 'int
Test::getValue()' discards qualifiers

```
// Demonstration of constant object,  
// show that constant object can only  
// call const member function  
#include<iostream>  
using namespace std;  
class Demo  
{  
    int value;  
public:  
    Demo(int v = 0) {value = v;}  
    void showMessage()  
    {  
        cout<<"inside showmessage()"<<endl;  
    }  
    void display() const  
    {  
        cout<<"inside display()"<<endl;  
    }  
};  
int main()  
{  
    //Constant object are initialised at the time of declaration using constructor  
    const Demo d1;  
    //d1.showMessage();Error occurred if uncomment.  
    d1.display();  
    return(0);  
}
```

OUTPUT : inside display()