

# Objektno orijentirano programiranje

## Vježba 1

### UNOS I ISPIS

Unos se realizira pomoću cin objekta klase istream, a ispis pomoću cout koji je objekt klase ostream. Objekt cout koristi se u kombinaciji s operatorom <<, a cin s operatorom >>.

Prepišite program bez kopiranja i testirajte ga.

```
#include <iostream>
int main()
{
    int a, b, s;
    std::cout << "unesi dva broja" << std::endl;
    std::cin >> a >> b;
    s = a + b;
    std::cout << "suma je " << s << std::endl;
}
```

### MANIPULATORI

Manipulatori se koriste za formatiranje ulaza i izlaza. Detaljnija tablica <http://www.cplusplus.com/reference/Library/manipulators/>

independent flags (switch on)	
boolalpha	ispis bool vrijednosti kao string
uppercase	velika slova u hex prikazu, E u eksponencijalnom zapisu
independent flags (switch off)	
noboolalpha	ispis bool vrijednosti kao broj 1 ili 0
nouppercase	ne generira velika slova u ispisu
numerical base format flags	
dec	decimalni format za cijele brojeve (baza 10)
oct	oktalni format za cijele brojeve (baza 8)
hex	heksadecimalni format za cijele brojeve (baza 16)
floating point format flags	
fixed	zapis sa pomičnim zarezom za realne brojeve
scientific	eksponencijalni zapis za realne brojeve
adjustment format flags	
left	lijevo poravnanje
right	desno poravnanje
input manipulatori	
ws	preskače početne whitespaceove
output manipulatori	
endl	kraj linije i flush
ends	ubacuje '\0' (null terminator) u izlaz
flush	prazni stream
parametrizirani manipulatori (<iomanip>)	
setprecision	preciznost decimalnih brojeva
setbase	zapis u bazi
setfill	znak za ispunjavanje
setiosflags	postavlja određene formate
resetiosflags	resetira određene formate

## ZADACI

1. Unesite sljedeći program i modificirajte ga testirajući gornje manipulatore.

```
#include <iostream>
#include <iomanip>
int main()
{
    bool flag;
    std::cout << "unesi " << true << " - " << false << std::endl;
    std::cin >> flag;
    std::cout << std::boolalpha << flag << std::endl;

    int a = 255;
    std::cout << "hex " << std::hex << a << std::endl;
    std::cout << "dec " << std::dec << a << std::endl;
    std::cout << "oct " << std::oct << a << std::endl;

    double pi = 3.141592;
    std::cout << "pi = " << std::scientific << std::uppercase;
    std::cout << std::setprecision(7) << std::setw(20);
    std::cout << std::setfill('0');
    std::cout << pi << std::endl;
}
```

2. Napišite dvije funkcije istog imena (preopterećenje za različite tipove, npr. `int` i `double`) koje ograničavaju vrijednost na interval `[low, high]`. Možete pretpostaviti da je `low <= high`.

Ako je vrijednost manja od `low`, funkcija vraća `low`; ako je veća od `high`, vraća `high`; inače vraća izvornu vrijednost.

Za `low` i `high` postavite pretpostavljene vrijednosti. U `main` pozovite obje varijante za nekoliko ulaza i ispišite rezultate.

3. Učitajte jedan redak u `std::string s` (`std::getline`). Koristeći range-based `for` sva slova pretvorite u velika, znamenke zamijenite znakom `'*'`, a razmake i tabove znakom `'_'`. Ispišite dobiveni string.

Napomena: za `std::string` koristite `#include <string>`, a za `std::toupper` koristite `#include <cctype>`

4. Napišite funkciju `at` koja vraća referencu na neki element C-niza. Koristeći povratnu vrijednost funkcije kao `lvalue` uvećajte `i`-ti element niza za jedan.