

[Sign up](#)[Code](#)[Issues](#) 0[Pull requests](#) 0[Projects](#) 0[Actions](#)[Wiki](#)[Security](#)[Pulse](#)

Branch: master ▾

[Find file](#)[Copy path](#)[cpp_kursu_odevleri](#) / [function_overloading](#) / [fo_06.md](#)

Fetching contributors...



111 lines (87 sloc) | 1.1 KB

[Raw](#)[Blame](#)[History](#)

Aşağıdaki kodlardan herbirinde function overloading ("işlev yüklemesi") olup olmadığını belirtin. Her bir kod için açıklama yapın:

```
void func(int);

void foo()
{
    extern func(int, int);
}
```

```
int func(double);
double func(double);
```

```
int f(double);
int f(double = 0.);
```

```
int g(int);
int g(const int);
```

```
int g(int *p);
int g(int* const p);
```

```
int f(int *p);
int f(const int *p);
```

```
void func(int &);
void func(const int &);
```

```
void g(int);  
void g(int &);
```

```
void foo(int &);  
void foo(int &&);
```

```
void f(const double &);  
void f(double &&);
```

```
typedef int Word;  
  
void f(Word);  
void f(int);
```

```
enum Color {gray, red, blue};  
void g(Color);  
void g(int);
```

```
#include <cstdint>  
  
void func(int32_t);  
void func(int);
```

```
int f(bool);  
int f(int);
```

```
void foo(char16_t);  
void foo(int);
```

```
void func(char);  
void func(signed char);
```

```
void f(char);  
void f(unsigned char);
```

```
void foo(int*)(int);  
void foo(int&)(int);
```

```
void g(int[10]);  
void g(int *p);
```

```
void foo(int(&)[10]);  
void foo(int(&)[20]);
```

© 2020 GitHub, Inc.

- [Terms](#)
- [Privacy](#)
- [Security](#)
- [Status](#)
- [Help](#)

- [Contact GitHub](#)
- [Pricing](#)
- [API](#)
- [Training](#)
- [Blog](#)
- [About](#)