**2023 08 09**

**Function Overloading**

***/\****

***function overloading***

***- derleme zamanında gerçekleşir.***

***a) compile-time***

***static binding***

***early binding***

***b) run time***

***dynaic binding***

***late binding***

***Bir fonksiyonun imzası (signature) fonksiyonun parametre değişkenlerinin türü ve sayısıdır***

***\*/***

***//redeclaration, overloading değil***

**int** foo**(int,int);**

**int** foo**(int,int);**

***//redeclaration, overloading değil***

**int** foo**(int);**

**int** foo**(const** **int);**

***// overloading***

**void** func**(int** **\***ptr**);**

**void** func**(const** **int** **\***ptr**);**

***//redeclaration***

**void** func**(int\*** ptr**);**

**void** func**(int\*** **const** ptr**);**

***// syntax hatası***

**int** foo**(int,int);**

**double** foo**(int,int);**

***// ikiside pointer (redeclaration)***

**void** func**(int** **\*);**

**void** func**(int** **[]);**

***// array decay (redeclaration)***

**void** func**(int(int));**

**void** func**(int\*(int);**

***// overloading dizilerin boyutları farklı***

**void** func**(int(\*)[**10**]);**

**void** func**(int(\*)[**12**]);**

***// redeclaration***

**void** func**(int(\*)[**10**])**

**void** func**(int[][**10**]);**

***// redeclaration***

**typedef** **int** Nec**;**

**void** func**(int);**

**void** func**(**Nec**);**

***// 3 tane overload var***

**void** func**(char);**

**void** func**(signed** **char);**

**void** func**(unsigned** **char);**

***// overloading var***

**void** func**(int);**

**void** func**(int** **&);**

***// 4 tane overload var***

**void** func**(int** **&)**

**void** func**(int** **&&)**

**void** func**(const** **int** **&)**

**void** func**(const** **int** **&&)**

***// overloading***

**void** foo**(int** **\*)**

**void** foo**(int** **&)**

***// overloading***

**void** foo**(int** **\*)**

**void** foo**(**std**::**nullptr\_t**)**

**void** func**(long** **double);**

**void** func**(char);**

**int** main**()**

**{**

***// syntax hatası çünkü iki fonksiyonda geçerli hangisi çağırılacak bilemeyiz.***

func**(**12.5**);** ***// ambiguity***

**}**

Overloading Süreci:

1. aday fonksiyonların saptanması
2. legal olarak çağıralabilecek fonksiyonlar tespiti

**void** func**(void** **\*);**

**void** func**(int** **\*);**

**void** func**(int,** **int);**

***// viable(uygun) fonksiyon yok ama 3 tane overload var***

**int** main**(){**

func**(**56**);** ***// amb***

**}**

***user defined conversion***

***standart conversion:***

***1) exact match (tam uyum)***

***2) promation (terfi - yükseltme)***

***3) conversion (dönüşüm)***

***//exact match spec:***

**void** func**(const** **int** **\*);**

**void** func**(int** **\*)**

**int** main**()**

**{**

**const** **int** x **=** 5**;**

func**(&**x**);** ***// const int \****

**int** y **=** 1**;**

***/\****

***eğer const ve const olmayan iki fonksiyonda tanımlıysa const olmayan***

***fonksiyon çağırılacak ama const olmayan tanımlı değilse bu sefer***

***const int \* çağırılacak***

***\*/***

func**(&**y**);**

**}**

**void** func**(int** **(\*)(int));**

**int** foo**(int);**

**int** main**(){**

**int** x **=** 10**;**

func**(&**x**);** ***//exact match***

**int** a**[**5**]{};**

func**(**a**);** ***//array decay (exact match)***

func**(&**foo**);** ***// int(int)***

**}**

**void** func**(int** **\*)**

**void** func**(double** **\*)**

**void** func**(**std**::**nullptr\_t**)**

**int** main**(){**

func**(nullptr)** ***// std::nullptr gelecek***

**}**

**class** MyClass**{};**

**void** func**(const** Myclass**&);** ***// const l value ref parametreli***

**void** func**(**Myclass **&&);** ***// r value ref parametreli***