* Signature
  + اصطلاحا به پارامترها یا آرگومان‌های **ورودی** یک تابع Signature‌ اطلاق می‌شود.
    - دقت کنید! فقط به پارامترهای ورودی! نه! ورودی و خروجی!
* Method Overloading
  + در هر کلاس، می‌توان از توابعی استفاده کرد که دارای نام یکسان بوده ولی دارای Signature‌ های متفاوتی می‌باشند.

با توجه به حضور تابع ذیل، بگویید که نوشتن کدام یک از توابع ذیل مجاز می‌باشد:

**int SomeFunction(int x, int y)  
{  
}**

int SomeFunction(int x, int y, int z)  
{  
}

**int SomeFunction(int x, int y)  
{  
}**

int SomeFunction(float x, float y)  
{  
}

**int SomeFunction(int x, int y)  
{  
}**

int SomeFunction(int x, float y)  
{  
}

**int SomeFunction(int x, int y)  
{  
}**

float SomeFunction(int x, int y)  
{  
}

**int SomeFunction(int x, int y)  
{  
}**

float SomeFunction(float x, float y)  
{  
}

**برداشت:** این که صرفا خروجی دو تابع هم نام و هم Signature متفاوت باشد، به معنای Overloading نیست، ولی ممکن است Overloading داشته باشیم، ولی خروجی‌های دو تابع نیز فرق داشته باشند.

**int SomeFunction(int x, int y)  
{  
}**

int SomeFunction(int A, int B)  
{  
}

**Access Modifiers**

**Class Access Modifiers**

public

private

**internal** (Default)

protected

protected internal

**Class Members**

Field

Property

Method

Event

**Class Members Access Modifiers**

public

**private** (Default)

protected

internal

protected internal

**Encapsulation**

**تعریف نادقیق:** هرگاه از کلاسی، متغیری تعریف کنیم، پس از زدن نقطه، بعد از نام آن متغیر، صرفا به تمام Member های public آن دسترسی خواهیم داشت.

**استاندارد قدیمی**

public class Person  
{  
 private string **\_fullName**;  
 public or protected or internal or protected internal string **FullName**;  
  
 public void SomeFunction(string **fullName**)  
 {  
 string **strFullName**;  
 }  
}

**استاندارد جدید**

**نام کلاس و نام Namespace و نام Method (Function) در هر حالت باید Pascal Case باشد!**

public class Person  
{  
 private string ـ**fullName**;  
 public or protected or internal or protected internal string **FullName**;  
  
 public void SomeFunction(string **fullName**)  
 {  
 string **fullName**;

string **fullNameString**;

}  
}