

Creating meaningful names

It is important to name a variable meaningfully. A computer can understand any code, but we have to write code for human understanding too. Here are some bad and good practices to be followed while naming a variable, a function and a class.

Bad Practice:

```
class SumSquare{
    fun add():Int{
        var s:Int =0
        for(number in 1 ≤ .. ≤ 100)
        {
            s += (number*number)
        }
        return s
    }
    fun square():Int{
        var pro :Int = (1 ≤ .. ≤ 100).sum() * (1 ≤ .. ≤ 100).sum()
        return pro
    }
}

fun main()
{
    val obj = SumSquare()
    println(obj.square() - obj.add())
}
```

- Having a single character as a variable name will not be understood by the co-developers.
- People can easily misunderstand the purpose of the function if we named it meaninglessly.

Good Practice:

```
class SumSquareDifference {  
    fun sumOfSquare():Int{  
        var sum:Int =0  
        for(number in 1..100)  
        {  
            sum += (number*number)  
        }  
        return sum  
    }  
    fun squareOfSum():Int{  
        var product :Int = (1..100).sum() * (1..100).sum()  
        return product  
    }  
}  
  
fun main()  
{  
    val sumSquareDifference = SumSquareDifference()  
    println(sumSquareDifference.squareOfSum() - sumSquareDifference.sumOfSquare())  
}
```

Variable Name :

- A variable name must tell the purpose of the variable.

Bad Practice:

```
var s:Int =0
```

Good Practice:

```
var sum:Int =0
```

Bad Practice:

```
var pro :Int = (1..100).sum() * (1..100).sum()
```

Good Practice:

```
var product :Int = (1..100).sum() * (1..100).sum()
```

Function Name:

- A function name must tell what action is performed by that function.

Bad Practice:

```
fun add():Int{  
    var s:Int =0  
    for(number in 1 ≤ .. ≤ 100)  
    {  
        s += (number*number)  
    }  
    return s  
}
```

Good Practice:

```
fun sumOfSquare():Int{  
    var sum:Int =0  
    for(number in 1 ≤ .. ≤ 100)  
    {  
        sum += (number*number)  
    }  
    return sum  
}
```

Bad Practice:

```
fun square():Int{  
    var pro :Int = (1 ≤ .. ≤ 100).sum() * (1 ≤ .. ≤ 100).sum()  
    return pro  
}
```

Good Practice:

```
fun squareOfSum():Int{  
    var product :Int = (1 ≤ .. ≤ 100).sum() * (1 ≤ .. ≤ 100).sum()  
    return product  
}
```

Class Name:

- A class name must tell about the purpose of the class.

Bad Practice:

```
class SumSquare{
```

Good Practice:

```
class SumSquareDifference {
```