

# Extractors

# Introduction

- ❖ Extractor is an object that has the method `unapply` as one of its members.

# The `apply` Method

- ❖ When an instance of a class is followed by parentheses with a list of zero or more parameters, the compiler invokes the `apply` method on that instance.
- ❖ We can define `apply` both in objects and in classes.

# The `unapply` Method

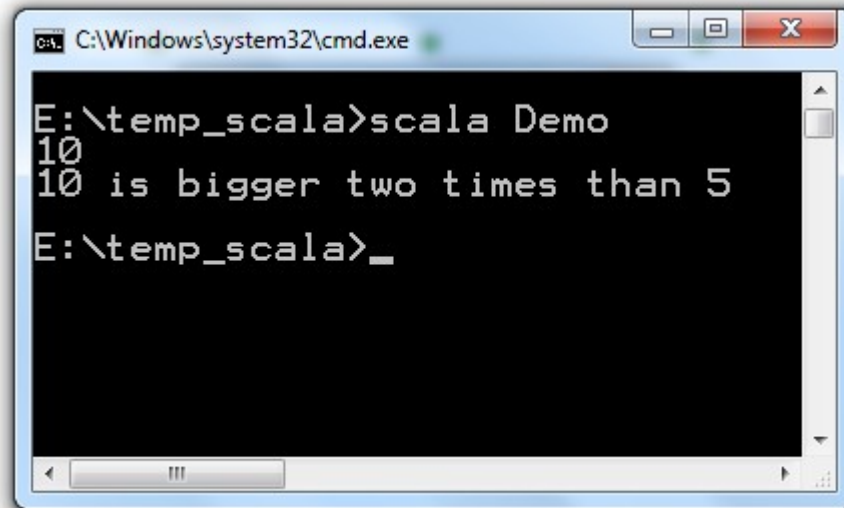
- ❖ The purpose of the `unapply` method is to extract a specific value we are looking for. It does the opposite operation `apply` does.
- ❖ When comparing an extractor object using the `match` statement the `unapply` method will be automatically executed.

# Sample

```
object Calculation
{
  def apply(x: Int) = x*2
  def unapply(z: Int): Option[Int] = if (z%2==0) Some(z/2) else None
}

object Demo
{
  def main(array: Array[String])
  {
    val x = Calculation(5)
    //apply is invoked... the value 10 is returned
    println(x)
    x match
    {
      case Calculation(num) => println(x+" is bigger two times than "+num)
      //unapply is invoked
      case _ => println("i cannot calculate")
    }
  }
}
```

# Sample



```
C:\Windows\system32\cmd.exe
E:\temp_scala>scala Demo
10
10 is bigger two times than 5
E:\temp_scala>_
```