PEP Scala (3)

Email: christian.urban at kcl.ac.uk
Slides & Code: KEATS

Office Hour: Fridays 13:00 – 14:00
Location: N7.07 (North Wing, Bush House)

Pollev: https://pollev.com/cfltutoratki576

Default Arguments

```
def collatzHelper(n: Int, a: Int = 0) : Int = ...


collatzHelper(n, 3)
collatzHelper(n, 0)

collatzHelper(n)   // a = 0
```

Last Week: Options & HO Funs.

```
List(7,2,3,4,5,6).find(_ < 4)
res: Option[Int] = Some(2)
```

```
List(5,6,7,8,9).find(_ < 4)
res: Option[Int] = None
```

```
List(1,2,3,4,5).map(x => x * x)
res: List[Int] = List(1, 4, 9, 16, 25)
```

Web-Crawler (1)

```
def get_page(url: String) : String = {
Try(fromURL(url)(”ISO-8859-1”).take(10000).mkString)
   .getOrElse { println(s” Problem with: $url”); ””}
}
```

Web-Crawler (2)

```
val http_pattern = """https?://[\^"]*""".r
val email_pattern =
 """([a-z\d\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})""".r


def unquote(s: String) = s.drop(1).dropRight(1)


def get_all_URLs(page: String): Set[String] =
  http_pattern.findAllIn(page).map(unquote).toSet

  // returns all URLs in a page
```

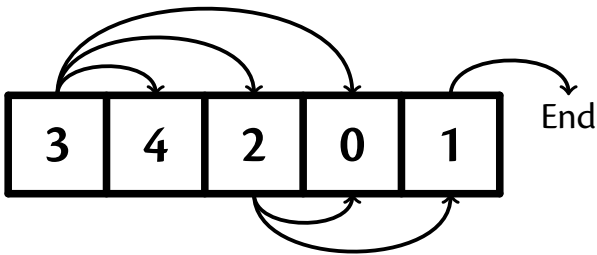
Web-Crawler (3)

```
def crawl(url: String, n: Int) : Unit = {
  if (n == 0) ()
  else {
    println(s"  Visiting: $n $url")
    val page = get_page(url)
    for (u <- get_all_URLs(page))
      crawl(u, n - 1)
  }
}
```

Email Harvester

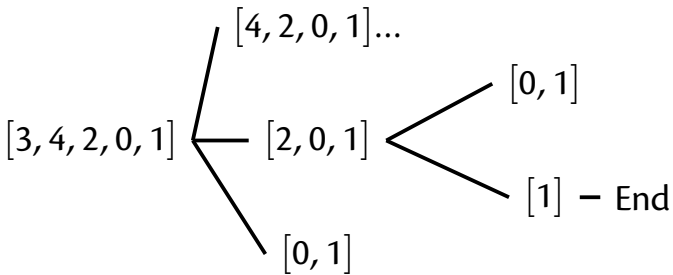
```
def emails(url: String, n: Int) : Set[String] = {
 if (n == 0) Set()
 else {
  println(s"  Visiting: $n $url")
  val page = get_page(url)
  val new_emails =
    email_pattern.findAllIn(page).toSet
  new_emails ++
    (for (u <- get_all_URLs(page))
       yield emails(u, n - 1)).flatten
 }
}
```

Jumping Towers



shortest: $3 \rightarrow 4 \rightarrow \text{End}$

next moves



Reverse Polish Notation

$$(3 + 1) * (2 + 9)$$

$$\Rightarrow$$

$$3 \;\; 1 \;\; + \;\; 2 \;\; 9 \;\; + \;\; *$$

Reverse Polish Notation

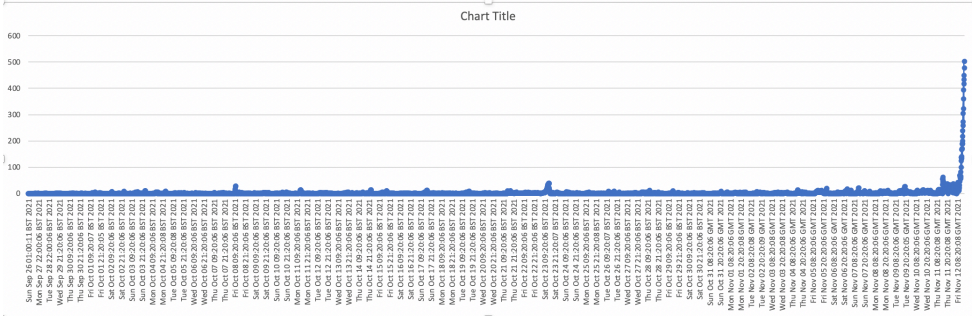
$$(3 + 1) * (2 + 9)$$

$$\Rightarrow$$

$$3\ \ 1\ +\ \ 2\ \ 9\ +\ *$$

```
ldc 3
ldc 1
iadd
ldc 2
ldc 9
iadd
imul
```

Testing Server



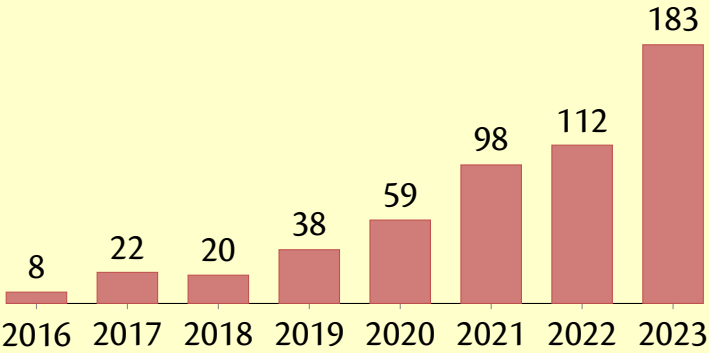
Feedback in CFL!

End-of-year feedback for 6CCS3CFL in 2019

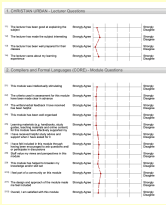
Unequivocally the worst module I've taken on this course. The subject matter is fascinating, however the insistence on the use of this abomination of a language "Scala" completely ruins it. If you're going to teach something as complex as this, use a proper language, not some "object oriented functional" abomination. Use C, you know, the language that real compilers are written in. I will go to the end of the earth to dissuade others from taking this module so long as Scala is still being used.

– Lone voice in the end-of-year feedback in 2019

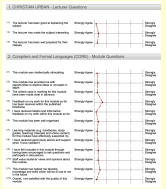
Students in CFL



Student numbers since the start of the compiler module.



2021



2022

One comment from this year

I feel like the module's point is to help us experience what it is like to program very challenging problems, it's not very realistic as in a realistic scenario we would have access to the internet, and other people's code and may collaborate. I feel like the point of the module is taken away due to how the plagiarism and collusion rules are put into place.

Another comment from this year

To prepare students for the C++ coursework better, for example introducing recursion and/or backtracking, because that is a big part of the coursework but wasn't even touched upon in the videos

Even another comment from this year

The coursework is too difficult.

- we reduced the amount of work this year and gave more time for C++ CW
- we recruited TA's for

  installation problems:
  - Oscar Sjostedt (`oscar.sjostedt@kcl.ac.uk`)
  - Nicole Lehchevska (`nicole.lehchevska@kcl.ac.uk`)

  github problems:
  - Quan Tran (`anh.tran@kcl.ac.uk`)

  discussion forum / general problems:
  - Ruben Ticehurst-James (`ruben.ticehurst-james@kcl.ac.uk`)

- we reduced the amount of work this year and gave more time for C++ CW
- we recruited TA's for

  installation problems:
  - Oscar Sjostedt (`oscar.sjostedt@kcl.ac.uk`)
  - Nicole Lehchevska (`nicole.lehchevska@kcl.ac.uk`)

  github problems:
  - Quan Tran (`anh.tran@kcl.ac.uk`)

  discussion forum / general problems:
  - Ruben Ticehurst-James (`ruben.ticehurst-james@kcl.ac.uk`)

Could you please spend the next 10 mins to fill out the end-of-year feedback. 🙏