6CCS3CFL – COURSEWORK 1

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_OPTIONAL QUESTIONS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Question 1:**

Email: [ayan.ahmad@kcl.ac.uk](mailto:ayan.ahmad@kcl.ac.uk)   
I am currently studying in London and will be studying on campus, in person face to face teaching.

**Question 2:**

I am a web developer, I started off with basic HTML and JS and CSS. Then I gained some experience in Adobe ColdFusion. I have made a few large business applications in PHP. I tend to focus more on the backend. Although I am extremely comfortable with frontend too.

I have decent experience in JavaScript (ReactJS and NodeJS) and am remarkably familiar with MySQL and MongoDB. I also have a decent amount of experience in Python, Java, C++, Flutter and Swift.

Through the courses at KCL, I have also worked in languages like: PDDL, Haskell, Ruby on Rails, Prolog, and a few more

I am currently learning Scala through this course and am thoroughly enjoying it. Lastly, I am learning Go for my own projects.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MANDATORY QUESTIONS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Question 3:**

|  |
| --- |
| **Nullable Expressions** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| **Der Expression** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Question 4**

Definitions:  
In order to implement CFUN, I have added some functions for CHAR, RANGE and ALL to translate their outputs to Booleans.

Code:

case class CFUN(f: Char => Boolean) extends Rexp

/\* CFUN translations \*/

def CHAR(c : Char) = CFUN((ch : Char) => c == ch)

def RANGE(s: Set[Char]) = CFUN((ch : Char) => s.contains(ch))

def ALL = CFUN ((\_ : Char) => true)

/\* End CFUN translations \*/

**Question 5**

Email: [ayan.ahmad@kcl.ac.uk](mailto:ayan.ahmad@kcl.ac.uk)

Regular Expression without running and simplifications:

SEQ(PLUS(RANGE(('a' to 'z').toSet ++ ('0' to '9').toSet + '\_' + '.' + '-')), SEQ(CHAR('@'), SEQ(PLUS(RANGE(('a' to 'z').toSet ++ ('0' to 9').toSet + '.' + '-')), SEQ(CHAR('.'), BETWEEN(RANGE(('a' to 'z').toSet + '.'),2,6)))))

Refer to attached code for implementation and methodology.

Regular Expression after ders and after applying simplifications:

**Question 6:**

|  |  |
| --- | --- |
| "/\*\*/" | yes |
| "/\*foobar\*/" | yes |
| "/\*test\*/test\*/" | no |
| "/\*test/\*test\*/" | yes |

**Question 7:**

R1 = (r1+)+

R2 = (r2+)+

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| “aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa” | |  |  | | --- | --- | | R1 | R2 | | Yes | Yes | |
| “aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa” | |  |  | | --- | --- | | R1 | R2 | | No | No | |
| “aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa” | |  |  | | --- | --- | | R1 | R2 | | No | Yes | |