Section 1:
Intro & RegEx

John, Robert, Rachel, Randy

Adapted from 20au, 22au

Announcements

- Due early next week (likely Tuesday): Project partner info
  - Check Ed discussion board if you need a partner!

March				
Monday	Tuesday	Wednesday	Thursday	Friday
14:30-15:20 Lecture 27 CSE2 G10 *Course introduction* slides	28	14:30-15:20 Lecture 29 CSE2 G10 *TBD*	Section 30 *TBD*	14:30-15:20 Lecture 31 CSE2 G10 *TBD*

WE ARE HERE

(this screenshot was taken when the calendar wasn't filled out yet.)

TAs' Tips

- Get started on project parts early!
  - *Especially* on the parts of the project which we explicitly advise to do so.
- Work *with* your partner on the project so you can both be up-to-speed on everything instead of dividing the tasks up amongst yourselves.
  - The project is entirely cumulative, so making sure that you both know all content helps with debugging the project as it progresses.
  - Project content can (*and likely will*) show up on exams as well, so make sure you know all of it.
- Instead of thrashing when you can't get a solution, come to OH!
- If you are having non-academic issues which are preventing you from meeting any deadlines (ex: illness), please contact the course staff ASAP so we can try to arrange an accommodation for you.

Icebreakers

- Find a partner, discuss what you did over Spring Break.
- Once time's up, state the following:
  - Your partner's name
  - Your partner's year
  - What you're partner is studying (major/ minor/ field(s) of interest/ research project)
  - What your partner did over Spring Break.

RegEx Worksheets!

Answers

Problem 1i

1) Describe the meaning of each of the following regular expressions in English and give two different strings it can produce:

i) (1 | 0)* 0

Problem 1i

1) Describe the meaning of each of the following regular expressions in English and give two different strings it can produce:

i) (1 | 0)* 0
Non-empty binary strings ending with 0

Problem 1ii

1) Describe the meaning of each of the following regular expressions in English and give two different strings it can produce:

ii) ([A-Z][a-z]* | [0-9]+)

Problem 1ii

1) Describe the meaning of each of the following regular expressions in English and give two different strings it can produce:

ii) ([A-Z][a-z]* | [0-9]+)
Sequence of lower case letters with first letter upper cased or sequence of base 10 digits

Problem 1iii

1) Describe the meaning of each of the following regular expressions in English and give two different strings it can produce:

iii) (ε | 4?0+1* X 3+)

Problem 1iii

1) Describe the meaning of each of the following regular expressions in English and give two different strings it can produce:

iii) (ε | 4?0+1* X 3+)
This one does not have a simple description, but some strings generated include ε, 0X3, 401X333, 40000111X333

Problem 2i

2) Write a regular expression for each of the following specifications:

i) All strings consisting of 0's and 1's (binary digits) with an even number of 0s

Problem 2i

2) Write a regular expression for each of the following specifications:

i) All strings consisting of 0's and 1's (binary digits) with an even number of 0s
1* (0 1* 0 1*)*

Problem 2ii

2) Write a regular expression for each of the following specifications:

ii) camelCased variable name in Java, where the alphabet is upper and lower-cased letters without any numbers or underscores

Problem 2ii

2) Write a regular expression for each of the following specifications:

ii) camelCased variable name in Java, where the alphabet is upper and lower-cased letters without any numbers or underscores
[a-z]+([A-Z][a-z]*)*

Note: this solution allows multiple upper-case letters to appear adjacent to each other.
Challenge: produce a solution that does not allow adjacent upper-case letters.

Problem 2iii

2) Write a regular expression for each of the following specifications:

iii) Non-empty binary strings where each 1 directly follows a 0 (challenge: only use symbols in table up until *)

Problem 2iii

2) Write a regular expression for each of the following specifications:

iii) Non-empty binary strings where each 1 directly follows a 0 (challenge: only use symbols in table up until *)

Challenge 1: (0 | 01) (0 | 01)*
Challenge 2 (no or): 0 (0* (0 1))* 0*
Normal: (0+1?)+