

CPSC 121,  
2007/8 Winter Term 2  
Sample Quiz 3

Name: \_\_\_\_\_

Student ID: \_\_\_\_\_

Signature: \_\_\_\_\_

- **You may want to complete what you consider to be the easiest questions first!**
- Justify all of your answers.
- No notes or electronic equipment are allowed.
- Keep your answers short. If you run out of space for a question, you have written too much.
- The number in square brackets to the left of the question number indicates the number of marks allocated to that question.
- Use the attached blank page for your rough work.
- Good luck!

Question	Marks
1	
2	
3	
Total	

UNIVERSITY REGULATIONS:

- Each candidate should be prepared to produce, upon request, his/her university-issued ID.
- No candidate shall be permitted to enter the examination room after the expiration of one half hour or to leave during the first half hour of the examination.
- **CAUTION:** candidates guilty of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action.
  1. Having at the place of writing, or making use of, any books, papers or memoranda, electronic equipment, or other memory aid or communication devices, other than those authorised by the examiners.
  2. Speaking or communicating with other candidates.
  3. Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.
- Candidates must not destroy or mutilate any examination material; must hand in all examination papers; and must not take any examination material from the examination room without permission of the invigilator.

1. Your friend is writing a regular expression to match prices between \$775 and \$850, inclusive. So far, she has the regular expression:

```
\$(7[7-9][5-9]|8[0-5][0-9])
```

Unfortunately, this produces some errors on her test suite:

**Strings that should be accepted:**

\$795 (accepted)

\$792 (rejected)

\$843 (accepted)

**Strings that should be rejected:**

\$852 (accepted)

\$925 (rejected)

\$771 (rejected)

Her regular expression rejects \$792 (which should be accepted) and accepts \$852 (which should be rejected). Correct the errors in the regular expression. Your answer should match the intended strings and, in particular, should correctly accept or reject all of the strings in the test suite.

2. Prove the following for arbitrary sets  $A$ ,  $B$ , and  $C$ :  $A \cup B \subseteq C \rightarrow A - C = \{\}$ . Your proof may be in a combination of English and logic but make sure that it's clear what the steps of your proof are and how each step follows from the previous steps. Your proof must not rely on Venn Diagrams.

3. Consider the set  $A = \{\text{hello}, 3, \emptyset\}$ , its power set  $P(A)$ , and its Cartesian product with itself  $A \times A$ . For each of the following functions, give an example value from the domain and its corresponding image. Then, say whether the function is (**Y**) or is not (**N**) injective and/or surjective. The first row is filled in as an example.

The function $f$	Pre-image	Image	Injective? ( <b>Y</b> or <b>N</b> )	Surjective? ( <b>Y</b> or <b>N</b> )
$f : A \rightarrow A$ maps values onto themselves: $f(x) = x$ .	hello	hello	<b>Y</b>	<b>Y</b>
$f : A \rightarrow (A \cup P(A))$ maps values onto themselves: $f(x) = x$ .				
$f : A \times A \rightarrow A$ picks out the first element from a 2-tuple: $f(x, y) = x$ .				
$f : A \rightarrow P(A)$ maps elements into sets: $f(x) = \{x\}$ .				
$f : P(A) \rightarrow P(A)$ “flips” sets: $f(B) = A - B$				
$f$ takes 2-tuples of elements from $A$ and maps them into subsets of $A$ . ( <b>You</b> pick the particular function.)				
$f : \{\{T, F\} \times \{T, F\}\} \rightarrow \{1, 2, 3, 4\}$ maps truth values to numbers: $f(p, q) = \begin{cases} 1 & \text{if } p \wedge q \\ 2 & \text{if } \sim q \wedge p \\ 3 & \text{if } \sim p \wedge q \\ 4 & \text{if } \sim (p \vee q) \end{cases}$				

## Appendix: Regular Expression Reference

Character(s)	Description	Example
.	Matches any character	.at = {"aat", "bat", ...}
[ ]	Matches one character from those listed	[cbr]at = {"cat", "bat", "rat"}
[ - ]	Matches one character from the range of characters listed	[a-c]at = {"bat", "cat", "dat"}
[ ^ ]	Matches one character from those not listed	[^b-d]at = {"aat", "eat", "fat", ...}
	Matches one item from those separated by pipes	(ph meerk r)at = {"phat", "meerkat", "rat"}
*	Repeat the previous item 0 or many times	ca*t = {"ct", "cat", "caat", ...}
+	Repeat the previous item 1 or many times	ca+t = {"cat", "caat", ...}
?	Repeat the previous item 0 or 1 time	ca?t = {"ct", "cat"}
{#}	Repeat the previous item an exact number of times	ca{3}t = {"caaat"}
{#, }	Repeat the previous item a min. number of times	ca{2, }t = {"caat", "caaat", ...}
{#, #}	Repeat the previous item between a min. and max. number of times	ca{1,3}t = {"cat", "caat", "caaat"}

In addition, there are a few special character codes that come in handy:

Symbol	Description
\d	Equivalent to [0-9]: Matches any digit
\D	Equivalent to [^0-9]: Matches any non-digit
\s	Matches white space character
\S	Matches a non-white space character
\w	Equivalent to [a-zA-Z0-9_]: Matches a "word" character
\W	Equivalent to [^a-zA-Z0-9_]: Matches a non-word character