WEEK 4

TABLE OF CONTENTS

HOMEWORK

HOMEWORK

- Should be corrected faster next time
  - Please double check your scoring → late night math's doesn't always work that well

HOMEWORK

- Any Questions to Week 3?

USEFUL METHODS AND CLASSES FOR THIS WEEK

CHARS

- Char is a digit between 0 and 127
- Each char is mapped to a letter
- A string is comprised of multiple chars
- ‘A’ == 65
  - char c = 65 is equivalent to c = ‘A’

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
32	[space]	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	|
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	[backspace]

USEFUL STRING METHODS

```
String s = “Demo”;
s.charAt(2);     // ’m’      First Letter is Index 0
s.length();      // 4         Starts at 0 being an empty string
```

BINARY OPERATORS

- These are the operators for Java
  - Differ in DS and ERA

Funktion	Opeator	Beispiel
`bitweises und`	&	$1001_2 \,\&\, 1010_2 = 1000_2$
`bitweises oder`	|	$1001_2 \mid 1010_2 = 1011_2$
`bitweises not`	~	$\sim 1010_2 = 0101_2$
`bitweises xor` ($\oplus$)	^	$1001_2 \,\hat{}\, 1010_2 = 0011_2$

BINARY OPERATORS – CARLOS DS TRAINER

Semantik aussagenlogischer Formeln als Tabellen

Für den unären Junktor $\neg$ gilt:

F	$\neg F$
0	1
1	0

~F

Für die binären Junktoren $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\otimes$, $\bar{\wedge}$ und $\bar{\vee}$ gilt:

F & G F | G F^G

F	G	$F \wedge G$	$F \vee G$	$F \rightarrow G$	$F \leftrightarrow G$	$F \otimes G$	$F \bar{\wedge} G$	$F \bar{\vee} G$
0	0	0	0	1	1	0	1	1
0	1	0	1	1	0	1	1	0
1	0	0	1	0	0	1	1	0
1	1	1	1	1	1	0	0	0

BINARY OPERATORS – PRACTICE

- 0010 & 1111
- 0000 | 1100
- ~1010
- 0011 ^ 0110

BINARY OPERATORS – PRACTICE EXAMPLES

- 0010 & 1111 = 0010
- 0000 | 1100 = 1100
- ~1010 = 0101
- 0011 ^ 0110 = 0101

TUTOR TASKS

NUMBER BASE

- Binary Base 2
- Octal Base 8
- Decimal Base 10
- Hexadecimal Base 16

NUMBER BASE

What numbers are valid in:

- Binary

NUMBER BASE

What numbers are valid in:

- Binary (0 and 1)

NUMBER BASE

What numbers are valid in:

- Octal

NUMBER BASE

What numbers are valid in:

- Octal (0-7)

NUMBER BASE

What numbers are valid in:

- Hexadecimal

NUMBER BASE

What numbers are valid in:

- Hexadecimal (0 - F)

NUMBER BASE – APPROACH

- Draw a multiplication/addition table

	0	1
0	0	0
1	0	1

Binary Multiplication Table

NUMBER BASE – APPROACH

- How would a binary addition table look like?

NUMBER BASE – APPROACH

- Draw a Base 5 Addition and Multiplication Table

NUMBER BASE – APPROACH

- Draw a Base 5 Addition and Multiplication Table
  - 33 + 14 + 13 in Base 5
  - 22*3 in Base 5

NUMBER BASE – TASKS

- $323478_9 + 111202337_9$
- $10101011100_2 * 11000111_2$
- $120022_3 * 22210_3$

NUMBER BASE – TASKS

- $323478_9 + 111202337_9$
- $1010101100_2 * 11000111_2$
- $120022_3 * 22210_3$
- $c01dc0ffe_{16} * deadaffe_{16}$

Base 16 Multiplication

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	0	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	0	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	0	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	0	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	0	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	0	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	0	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	0	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	0	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	0	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

NUMBER BASE CONVERSION - DEMO

- 100 in Binary
- 1101 1111 0b in Hexadecimal
- 1101 1010 1111b in Decimal

NUMBER BASE CONVERSION – TUTOR TASKS

- 10101011100_2 in Base10
- 10101011100_2 in Base16
- 354347357_{10} in Base 2

NUMBER BASE – CONVERSION DEMO FOR HOMEWORK

- 2143 in base 5 to base 7
- 21432 in base 5 to base 9

- Double check your solutions using Wolfram Alpha "21432_5 in base 9"

CAESAR ENCRYPTION

CAESAR ENCRYPTION

- Only encrypts Letters, not symbols
- "Hello Students! .aAbBcC? >wWxXyYzZ<“ becomes "Khoor Vwxghqwv! .dDeEfF? >zZaAbBcC<“ with a shift of 3

CAESAR ENCRYPTION

- Input a String to be encrypted
- Input a cipher as an int
  - Can be negative or greater than 26
- Encrypt the string
- Case should remain the same
- Output the String via write()

VOWEL REPLACEMENT

VOWEL REPLACEMENT

- Write a program that replaces all vowels (a, e, i, o u) with a specified letter
  - Ä, ö, ü are not considered vowels
- Must keep capitalization
- Only uses length and charAt library functions

EX: "Exenmeister" to "Oxonmoostor" if O/o is inputted

VOWEL REPLACEMENT – APPROACH

- Use code interface provided
- Input a letter to replace all vowels with
- Output the new String

INVERSE CAPITALIZATION

INVERSE CAPITALIZATION

- Read String
- Swap Upper and Lower Case
- Outputs via Write

Only uses length and charAt library functions

EX: "Hello Students!" to "hELLO sTUDENTS!"

INVERSE CAPITALIZATION

- Challenge for the experienced programmers:
  - Convert uppercase to lowercase and vice versa via binary operators

- Tip: Look at the ASCII table in Binary and compare a letters uppercase and lowercase number