Character Data Type and Operations

Hui Chen

Department of Computer & Information Science

Brooklyn College

Objectives

- To represent characters using the **char** type (§4.3).

- To encode characters using ASCII and Unicode (§4.3.1).

- To represent special characters using the escape sequences (§4.4.2).

- To cast a numeric value to a character and cast a character to an integer (§4.3.3).

- To compare and test characters using the static methods in the **Character** class (§4.3.4).

Outline

- Discussed

  - Math class and its class/static methods and constants

- Characters

  - Character data type

  - The Character class

  - Casting between characters and integers

Character Data Type

- Examples

char letter = 'A'; (ASCII)

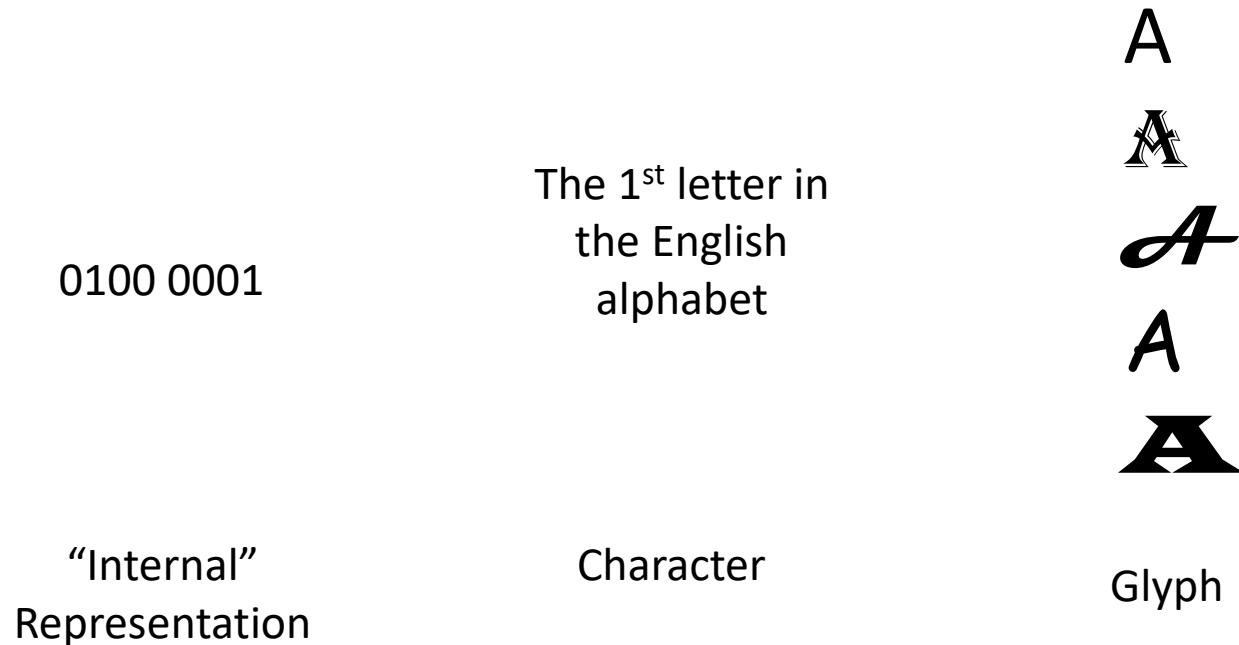char numChar = '4'; (ASCII)

Four hexadecimal digits.

char letter = '\u0041'; (Unicode)

char numChar = '\u0034'; (Unicode)

Character and Character Encoding

- But, what is a character? How does a computer system represent a character?

A

0100 0001

The 1st letter in the English alphabet

"Internal" Representation

Character

Glyph

Unicode and Java

- Recommended reading

  - https://docs.oracle.com/javase/tutorial/i18n/text/unicode.html
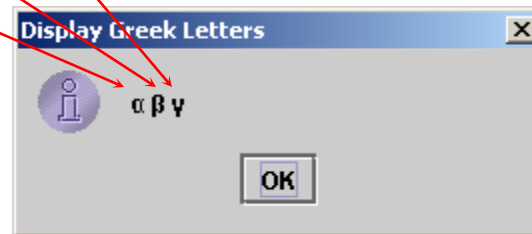
Unicode

- Java characters use *Unicode*, originally a 16-bit encoding scheme established by the Unicode Consortium to support the interchange, processing, and display of written texts in the world's diverse languages.

- 16-bit Unicode takes two bytes, preceded by \u, expressed in four hexadecimal numbers that run from '\u0000' to '\uFFFF'.

- So, 16-bit Unicode can represent `65535 + 1 characters`

- Because 16-bit encoding supports 2^{16} (65,536) characters, which is insufficient to define all characters in use throughout the world, the Unicode standard was extended to 0x10FFFF, which supports over one million characters.

Unicode Examples for the World's Languages

Unicode \u03b1 \u03b2 \u03b3 for three Greek letters

Increment and Decrement

- The increment and decrement operators can also be used on char variables to get the next or preceding Unicode character.

- For example, the following statements display character b.

  ```
  char ch = 'a';

  System.out.println(++ch);
  ```

ASCII Character Set

- ASCII Character Set is a subset of the Unicode from \u0000 to \u007f

TABLE B.1   ASCII Character Set in the Decimal Index

	0	1	2	3	4	5	6	7	8	9
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht
1	nl	vt	ff	cr	so	si	dle	dcl	dc2	dc3
2	dc4	nak	syn	etb	can	em	sub	esc	fs	gs
3	rs	us	sp	!	"	#	$	%	&	'
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	`	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{	|	}	~	del		

ASCII and Unicode

- What are the hexadecimal numbers of decimal numbers 48, 57, 65, 90, 97, and 122

Characters	Code Value in Decimal	Unicode Value
'0' to '9'	48 to 57	\u0030 to \u0039
'A' to 'Z'	65 to 90	\u0041 to \u005A
'a' to 'z'	97 to 122	\u0061 to \u007A

Escape Sequences for Special Characters

Escape Sequence	Name	Unicode Code	Decimal Value
\b	Backspace	\u0008	8
\t	Tab	\u0009	9
\n	Linefeed	\u000A	10
\f	Formfeed	\u000C	12
\r	Carriage Return	\u000D	13
\\	Backslash	\u005C	92
\"	Double Quote	\u0022	34

Casting between char and Numeric Types

- Examples

- `int i = 'a'; // Same as int i = (int)'a';`

- `char c = 97; // Same as char c = (char)97;`

Comparing and Testing Characters

- How about this method? This is in fact not a recommended method. Avoid it.

```
if (ch >= 'A' && ch <= 'Z')
  System.out.println(ch + " is an uppercase letter");
else if (ch >= 'a' && ch <= 'z')
  System.out.println(ch + " is a lowercase letter");
else if (ch >= '0' && ch <= '9')
  System.out.println(ch + " is a numeric character");
```

Methods in the Character Class

Method	Description
isDigit(ch)	Returns true if the specified character is a digit.
isLetter(ch)	Returns true if the specified character is a letter.
isLetterOfDigit(ch)	Returns true if the specified character is a letter or digit.
isLowerCase(ch)	Returns true if the specified character is a lowercase letter.
isUpperCase(ch)	Returns true if the specified character is an uppercase letter.
toLowerCase(ch)	Returns the lowercase of the specified character.
toUpperCase(ch)	Returns the uppercase of the specified character.

Questions?