

COMPSCI 210 S1T

Computer Systems

Alphanumeric Representation

ASCII & Unicode

Agenda & Reading

- ◆ Agenda:
 - ASCII
 - Unicode
 - Unicode in Java
 - UTF
 - Conversion
- ◆ Animation
 - 05ucs2utf.htm
 - 05utf2ucs.htm
- ◆ Examples
 - 05\UnicodeDisplay.jar
 - 05\UnicodeTableSample.jar
 - 05\PrintUnicodeApplet.html
 - 05\UnicodeConverter.html
- ◆ Exercise:
 - 05

COMPSCI210 - 04

2

Introduction

- ◆ Alphanumeric
 - Describing a code or string of characters that includes both letters and numbers and/or punctuation marks or mathematical symbols.
 - ◆ Letter
 - ◆ Digit
 - ◆ Period, hyphen, punctuation
 - ◆ Mathematic symbol
 - ◆ TAB, CR (carriage return), LF (new line)
 - ◆ etc
- ◆ Standardize code
 - ASCII Code – The American Standard Code for Information Interchange (proposed in 1963)

COMPSCI210 - 04

3

ASCII

- ◆ 128 decimal numbers ranging from 0 to 127
- ◆ 7 bits
- ◆ Consists of
 - Letters (upper and lower)
 - ◆ ASCII code for 'A' is 41 (hex).
 - ◆ ASCII code for 'a' is 61 (hex).
 - Numeric + punctuation
 - 32 transmission control and formatting characters,
 - ◆ TAB (Horizontal Tab), code = 09
 - ◆ LF (line feed/ new line), code = 0A
 - ◆ CR (carriage return), code = 0D
 - ◆ SP (space), code = 20
 - ◆ ...

COMPSCI210 - 04

4

Unicode

- ◆ English has 26 letters.
- ◆ How about other characters?
 - Like Greek, Chinese, Korean...
- ◆ How to represent them?
 - Not enough space!!
- ◆ UCS-2 (Universal Multiple-Octet Coded Character Set)
 - 16-bit character set
 - ◆ Provide code for more than 65,000 characters
 - ◆ Designed to cover all the world's major living languages, scientific symbols ...
- ◆ Java uses Unicode internally as the basis for its String class.

COMPSCI210 - 04

5

Examples

- ◆ A = 0041 (Hex)
- ◆ β = 03B2
- ◆ 4E00

一 丁 丂 七 丩 丅 丅 万 丈 三 上 下 丌 丌

- ◆ 3041

あ あ い い う う え え お お か が き ぎ く

UnicodeTableSample.jar

UnicodeDisplay.jar

COMPSCI210 - 04

More Examples:

[UnicodeDisplay.jar](#)
[UnicodeTableSample.jar](#)
[PrintUnicodeApplet.html](#)
[UnicodeConverter.html](#)

Unicode in Java

- ◆ char (2 bytes)

```
char c = 0x0041;  
Char c2 = 0x4E00;
```

- ◆ String

```
String s = "\u4f60\u597d\u55ce";
```

- ◆ NOTE: Use the font that supports Unicode characters

- E.g.
 - ◆ Arial Unicode MS
 - ◆ Lucida Sans Unicode

```
Font font = new Font("Arial Unicode MS", Font.BOLD, 18);  
g.setFont(font);  
s = "\u4f60\u597d\u55ce?";  
g.drawString(s, 50, 30);
```

COMPSCI210 - 04

7

UTF-8

- ◆ Unicode text can be represented in another format:
- ◆ Unicode Transformation Format (UTF-8)
- ◆ Transform Unicode characters to variable length encoding of bytes (8/16/24 bits)
- ◆ Must be:

- One-to-one mapping
- Unambiguous

- ◆ Advantages:

- Similar to ASCII
- Used with much existing software

	UCS-2	UTF-8
A	0041	41
	4E00	E4 B8 80
©	00A9	C2 A9

COMPSCI210 - 04

8

Convert from UCS-2 -> UTF-8

◆ Animation: 05ucs2utf.htm

- Case 1: (9 or more leading zeros)
 - ◆ 0000 0000 0xxx xxxx -> 0xxx xxxx -> ASCII code
 - ◆ Example: 0041(Hex)
- Case 2: (5 – 8 leading zeros)
 - ◆ 0000 0xxx xxyy yyyy to 0000 0000 xxyy yyyy
 - > 0xxx xxxx -> 110x xxxx 10yy yyyy
 - ◆ Example: 03B1 (Hex)
- Case 3: (4 or fewer leading zeros)
 - ◆ xxxx yyyy yyzz zzzz -> 0xxx xxxx
 - > 1110 xxxx 10yy yyyy 10zz zzzz
 - ◆ Example: 5E F6 (Hex)

Exercise

◆ Convert 0062 0073 0042 662E 0020 5DF6
from ucs-2 to utf-8

Convert UTF-8 to UCS-2

◆ Animation: 05utf2ucs.htm

- Case 1: starts with 0
 - > encoded in a single byte, equivalent to ASCII
 - ◆ Example: 20 (hex)
- Case 2: starts with 110
 - > Add 5 leading zeros + the last 5 bits of this byte + the last 6 bits of the next byte
 - ◆ Example: C7 (hex), take two bytes: C7 9A
- Case 3: starts with 1110
 - > the last 4 bits of this byte + 6 bits of the 2nd byte + 6 bits of the last byte
 - ◆ Example: E6 (hex), take 3 bytes: E6 98 AF

Exercise

◆ Convert 21 E8 A2 93 C7 8E from utf-8 to
ucs-2