

CS 113 – Computer Science I

Lecture 08 – Recursion, Strings, Arrays

Thursday 09/28/2023

Announcements

- HW02:
 - Due tonight

- HW03 releasing tonight
 - Due Monday 10/02
- Read & Follow Instructions
 - Don't just skim the labs & homework

Agenda

Recursion

Comparing strings

• In Java, you cannot directly compare strings using ==

- Instead, use **compareTo**
 - Javadocs: https://docs.oracle.com/javase/7/docs/api/java/lang/String.html

compareTo

```
public int compareTo(String anotherString)
```

Compares two strings lexicographically. The comparison is based on the Unicode value of each character in the strings. The character sequence represented by this String object is compared lexicographically to the character sequence represented by the argument string. The result is a negative integer if this String object lexicographically precedes the argument string. The result is a positive integer if this String object lexicographically follows the argument string. The result is zero if the strings are equal; compareTo returns 0 exactly when the equals (Object) method would return true.

This is the definition of lexicographic ordering. If two strings are different, then either they have different characters at some index that is a valid index for both strings, or their lengths are different, or both. If they have different characters at one or more index positions, let *k* be the smallest such index; then the string whose character at position *k* has the smaller value, as determined by using the < operator, lexicographically precedes the other string. In this case, compareTo returns the difference of the two character values at position *k* in the two string -- that is, the value:

```
this.charAt(k)-anotherString.charAt(k)
```

If there is no index position at which they differ, then the shorter string lexicographically precedes the longer string. In this case, compareTo returns the difference of the lengths of the strings -- that is, the value:

```
this.length()-anotherString.length()
```

Specified by:

compareTo in interface Comparable < String >

Parameters:

anotherString - the String to be compared.

Returns:

the value 0 if the argument string is equal to this string; a value less than 0 if this string is lexicographically less than the string argument; and a value greater than 0 if this string is lexicographically greater than the string argument.

09/28/23 CS 131 – Fall '23 - Lecture 08

public int compareTo(String anotherString)

Parameters:

anotherString - the String to be compared.

Returns:

- the value 0 if the argument string is equal to this string;
- a value less than 0 if this string is lexicographically less than the string argument;
- and a value greater than 0 if this string is lexicographically greater than the string argument.

Comparing strings

• In Java, you cannot directly compare strings: use **compareTo**

```
String a = "apple";
String b = "banana";
if (a.compareTo(b) == 0) {
    System.out.println("a and b match!");
}
if (a.compareTo(b) != 0) {
    System.out.println("a and b DO NOT match!");
}
```

Lexicographic Values/Order

- Strings are ordered lexicographically
 - Generally, the same order as alphabetical order, with some caveats
 - The characters of a string each correspond to a number

ASCII

0 0 000 NUL (null) 1 1 001 SOH (start of heading) 2 2 002 STX (start of text) 3 3 21 041 ! ! 3 66 41 101 A A 97 61 141 a a 2 2 002 STX (start of text) 3 4 22 042 " " 6 64 42 102 B B 98 62 142 b b 3 3 003 ETX (end of text) 3 5 23 043 # # 6 64 41 104 D D 100 64 144 d d 5 5 005 ENO (enquiry) 3 7 25 045 ! \$ 6 9 45 105 E E 100 64 144 d d 5 7 7 007 BEL (bell) 3 9 27 047 ' ' 7 1 47 107 G G 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 1 B 013 VT (vertical tab) 1 B 013 VT (vertical tab) 1 E 013 SOH (saminary return) 1 E 013 SOH (shift un) 1 F 015 CR (carriage return) 1 E 010 SOD DLE (data link escape) 1 7 10 C1 (device control 1) 1 10 21 DC1 (device control 2) 1 10 10 20 DLE (data link escape) 1 11 021 DC1 (device control 2) 1 12 DC3 MAK (negative acknowledge) 2 14 8 030 CAN (cancel) 3 2 0 040 Space 4 4 0 100 @ 0 4 40 100 @ 0 4 64 100 A A 9 76 1 141 a a 66 42 102 B B 8 66 42 102 B B 9 8 62 142 b b 66 42 102 B B 9 8 62 142 b b 67 43 103 C C 9 63 143 c c 67 43 103 C C 9 63 143 c c 100 64 144 d d 67 47 107 G G 100 66 145 g f 100 66 154 g f 100 66 155 i i 100 67 147 g g 100 67 144 d d 100 68 150 d d 100 64 144 d d 100 64 144 d d 100 66 145 g f 100 66 154 g f 100 66 155 g f 100 67 147 g g 100 68 150 i f 100 69 151 i f
2 2 002 STX (start of text) 34 22 042 %#34;" 66 42 102 %#66; B 98 62 142 %#98; b 3 3 003 ETX (end of text) 35 23 043 %#35; # 67 43 103 %#67; C 99 63 143 %#99; C 4 4 004 EOT (end of transmission) 36 24 044 %#36; \$ 68 44 104 %#68; D 100 64 144 %#100; d 5 5 005 ENQ (enquiry) 37 25 045 %#37; \$ 69 45 105 %#69; E 101 65 145 %#101; e 6 6 006 ACK (acknowledge) 38 26 046 %#38; \$ 70 46 106 %#70; F 102 66 146 %#102; f 7 7 007 BEL (bel1) 39 27 047 %#39; ' 71 47 107 %#71; G 103 67 147 %#103; G 8 8 010 BS (backspace) 40 28 050 %#40; (72 48 110 %#72; H 104 68 150 %#104; h 11 8 013 VT (vertical tab) 41 29 051 6#41;) 73 49 111 6#73; I 105 69 151 6#105; i 11 B 013 VT (vertical tab) 42 2A 052 6#42; * 74 4A 112 6#74; J 106 66 152 6#106; j 11 B 015 VT (vertical tab) 42 2A 052 6#42; * 74 4A 112 6#74; J 106 66 153 6#107; k 11 0 65 C (carriage return) 45 2D 055 6#46; - 78 4E 116 6#78; N 110 6E 156 6#110; n 15 F 017 SI (shift out) 46 2E 056 6#46; - 78 4E 116 6#78; N 110 6E 156 6#110; n 15 F 017 SI (shift in) 47 2F 057 6#47; / 79 4F 117 6#79; O 111 6F 157 6#111; O 16 10 020 DLE (data link escape) 49 31 061 6#49; 1 81 51 121 6#81; O 111 6F 157 6#111; O 16 10 020 DLE (device control 1) 49 31 061 6#49; 1 81 51 121 6#81; O 111 67 160 6#112; P 112 70 160 6#112; P 113 023 DC3 (device control 3) 51 33 063 6#51; 3 83 53 123 6#33; S 115 73 163 6#115; S 115 025 NAK (negative acknowledge) 53 35 065 6#53; 5 85 51 25 6#85; U 117 75 165 6#1119; W 24 18 030 CAN (cancel) 56 38 070 6#56; 8 8 88 130 6#88; X 120 78 170 6#120; X
3 3 003 ETX (end of text) 4 4 004 EOT (end of transmission) 5 5 005 ENQ (enquiry) 37 25 045 6#37; % 6 6 006 ACK (acknowledge) 38 26 046 6#38; 6 77 007 BEL (bell) 39 27 047 6#39; ' 70 46 106 6#70; F 70 70 BEL (bell) 39 27 047 6#39; ' 71 47 107 6#71; G 103 67 147 6#103; g 8 8 010 BS (backspace) 40 28 050 6#40; (72 48 110 6#72; H 104 68 150 6#104; h 12 0 14 FF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 35 23 043 6#35; # 67 43 103 6#67; C 99 63 143 6#99; C 68 44 104 6#68; D 100 64 144 6#100; d 67 43 103 6#67; C 99 63 143 6#99; C 68 44 104 6#68; D 100 64 144 6#101; d 68 44 104 6#68; D 100 64 144 6#102; f 70 46 106 6#70; F 102 66 146 6#102; f 70 47 107 6#71; G 103 67 147 6#103; g 70 47 12 107 6#71; G 103 67 147 6#103; g 71 47 107 6#71; G 103 67 147 6#103; g 72 48 110 6#72; H 73 49 111 6#72; H 74 4A 112 6#74; J 75 4B 113 6#75; K 107 6B 153 6#105; i 76 4C 114 6#76; L 77 4D 115 6#77; M 109 6D 155 6#109; m 78 4E 116 6#78; N 110 6E 156 6#110; n 78 4E 116 6#78; N 110 6E 156 6#110; n 78 4E 116 6#78; N 110 6E 156 6#110; n 78 4E 116 6#78; N 110 6E 156 6#110; n 78 4E 116 6#78; N 110 6E 156 6#110; n 78 4E 116 6#78; N 110 6E 156 6#110; n 78 4E 116 6#78; N 110 70 160 6#112; p 83 53 123 6#83; S 115 73 163 6#115; s 83 53 123 6#88; S 115 73 163 6#115; s 83 56 126 6#86; V 118 76 166 6#118; V 84 18 030 CAN (cancel) 85 57 07 6#855; 7 87 57 127 6#87; W 119 77 167 6#119; W 84 18 030 CAN (cancel) 85 60 66 6#56; 8 86 58 130 6#88; X 120 78 170 6#102;
4 4 004 EOT (end of transmission) 5 5 005 ENO (enquiry) 6 6 006 ACK (acknowledge) 7 7 007 BEL (bell) 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 24 04 04 4 6#36; \$ 36 24 044 6#36; \$ 37 25 045 6#37; \$ 46 24 044 6#36; \$ 38 26 046 6#38; \$ 38 26 046 6#38; \$ 39 27 047 6#39; \$ 71 47 107 6#71; \$ 69 45 105 6#69; E 101 65 145 6#101; e 102 6#102; f 103 67 147 6#103; g 103 67 147 6#103;
5 5 005 ENQ (enquiry) 6 6 6 006 ACK (acknowledge) 7 7 007 BEL (bell) 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 S0 (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 10 020 DLE (data link escape) 18 11 020 DC2 (device control 1) 19 13 023 DC3 (device control 2) 19 13 023 DC3 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 24 18 030 CAN (cancel) 37 25 045 6#37; % 38 26 046 6#38; 6 37 04 46 106 6#70; F 38 26 046 6#39; 1 39 27 047 6#39; 1 47 27 28 110 6#71; 0 104 68 150 6#105; 1 105 69 151 6#105; 1 105 69 151 6#105; 1 105 69 151 6#105; 1 105 69 151 6#105; 1 105 69 151 6#105; 1 107 6B 152 6#105; 1 107 6B 153 6#105; 1 107 6B 153 6#105; 1 108 6C 154 6#108; 1 107 6B 153 6#107; 1 109 6D 155 6#109; 1 109 6D 155 6#109; 1 109 6D 155 6#109; 1 110 6E 156 6#110; 1 110 6E 156 6#110; 1 110 6E 166 156 144 6#105; 1 110 6E 156 145 6#105; 1 110 6E 156 145 6#105; 1 110 6E 156 150; 1 110 6E 156 140; 1 110 6E 156 150; 1 110 6E 156 164; 1 110 6E 156 150; 1 110 6E 156 105; 1 110 6E 156 150; 1 110 6E 156 164; 1 110 6E 156 150; 1 110 6E 156 164; 1 110 6E
6 6 006 ACK (acknowledge) 7 7 007 BEL (bell) 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 10 10 21 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 24 18 030 CAN (cancel) 38 26 046 6#38; 6 39 27 047 6#39; ' 40 28 050 6#40; (41 29 051 6#41;) 40 28 050 6#40; (41 29 051 6#41;) 41 29 051 6#41;) 42 2A 052 6#42; * 44 2C 054 6#42; * 45 2D 055 6#45; - 46 2E 056 6#46; . 47 4D 115 6#77; M 109 6D 155 6#109; m 48 2E 056 6#46; . 48 4E 116 6#78; N 110 6E 156 6#110; n 49 31 061 6#49; 1 80 50 120 6#80; P 112 70 160 6#112; p 81 51 121 6#61; 0 81 51 121 6#61; 0 81 51 121 6#61; 0 81 51 124 6#64; T 116 74 164 6#113; Q 82 52 122 6#85; U 117 75 165 6#117; u 83 53 103 6#88; X 120 6#86; V 118 76 166 6#118; v 84 18 030 CAN (cancel) 85 38 070 6#56; 8 85 130 6#88; X 120 78 170 6#120; X
7 7 007 BEL (bell) 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 S0 (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 24 18 030 CAN (cancel) 39 27 047 6#39; ' 40 28 050 6#40; (40 28 050 6#40; (40 28 050 6#40; (41 29 051 6#41;) 42 2A 052 6#42; * 43 2B 053 6#43; + 44 2C 054 6#44; , 45 2D 055 6#43; + 46 2E 056 6#46; . 47 4D 115 6#77; M 109 6D 155 6#109; m 48 30 060 6#48; 0 48 30 060 6#48; 0 48 30 060 6#48; 0 48 30 060 6#48; 0 49 31 061 6#49; 1 49 31 061 6#49; 1 49 31 061 6#49; 1 49 31 061 6#49; 1 40 024 DC4 (device control 4) 40 30 066 6#54; 6 40 147 6#71; G 40 104 68 150 6#104; h 41 29 051 6#41;) 42 48 110 6#72; H 44 112 6#74; J 45 107 6B 153 6#107; k 46 2E 056 6#46; . 47 4D 115 6#77; M 40 115 6#105; i 41 02 64 CH (device control 1) 42 2B 055 6#44; , 43 2B 053 6#43; + 44 2C 054 6#46; , 76 4C 114 6#76; L 47 107 6B 153 6#107; k 48 2D 055 6#44; , 76 4C 114 6#76; L 48 116 6#78; N 49 111 6#73; I 40 66 A 152 6#106; j 40 76 4E 116 6#77; M 40 115 6#77; M 41 10 6E 156 6#110; n 47 2F 057 6#47; / 48 110 6#77; M 48 116 6#113 6#77; M 49 111 6#13 6#105; i 40 2B 055 6#45; - 40 2B 055 6#45; - 41 2 055 6#45; - 42 2B 055 6#45; - 43 2B 053 6#43; + 44 2C 054 6#46; . 77 4D 115 6#77; M 40 115 6#77; M 41 116 6#13 6#105; 1 41 11
8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SVM (synchronous idle) 24 18 030 CAN (cancel) 40 28 050 6#40; (41 29 051 6#41;) 42 2A 052 6#42; * 43 2B 053 6#43; + 44 2C 054 6#42; * 45 2D 055 6#45; - 46 2E 056 6#46; . 47 4A 112 6#74; J 106 6A 152 6#106; j 48 113 6#75; K 107 6B 153 6#107; k 48 116 6#76; L 108 6C 154 6#108; l 77 4D 115 6#77; M 109 6D 155 6#109; m 78 4E 116 6#78; N 109 6D 155 6#109; m 78 4E 116 6#78; N 109 6D 155 6#101; n 79 4F 117 6#79; O 111 6F 157 6#111; o 80 50 120 6#80; P 112 70 160 6#112; p 80 50 120 6#80; P 112 70 160 6#112; p 81 51 121 6#81; Q 113 71 161 6#113; q 82 52 122 6#82; R 114 72 162 6#114; r 83 53 123 6#83; S 115 73 163 6#115; S 84 54 124 6#84; T 116 74 164 6#116; t 84 16 6#78; N 109 6D 155 6#109; m 85 55 125 6#85; U 117 75 165 6#117; u 86 56 126 6#86; V 118 76 166 6#118; v 87 57 127 6#87; W 119 77 167 6#119; w 88 58 130 6#88; X 120 78 170 6#120; X
9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 23 17 027 ETB (end of trans. block) 24 2 A 052 c#42; * 44 2 C 054 c#42; * 45 2D 055 c#45; - 46 2E 056 c#46; . 47 2F 057 c#47; / 48 111 c#73; I 106 6A 152 c#106; J 75 4B 113 c#75; K 107 6B 153 c#107; k 108 6C 154 c#108; I 77 4D 115 c#77; M 109 6D 155 c#109; m 110 6E 156 c#110; n 15 F 017 SI (shift in) 47 2F 057 c#47; / 48 30 060 c#48; 0 80 50 120 c#80; P 112 70 160 c#112; p 81 51 121 c#81; Q 82 52 122 c#82; R 83 53 123 c#83; S 84 54 124 c#84; T 84 54 124 c#84; T 85 57 127 c#87; W 87 57 127 c#87; W 88 58 130 c#88; X 88 150 c#105; i 84 41;) 84 44 12 2 A 052 c#42; * 84 44 112 c#74; J 86 66 152 c#106; j 87 44 A 112 c#74; J 88 106 6 A 152 c#106; j 88 151 106 6A 152 c#106; j 88 151 106 6A 152 c#106; j 88 151 106 6A 152 c#106; j 88 151 107 6B 153 c#107; k 89 107 6B 153 c#106; j 89 107 6B 153 c#107; k 89 107 6B 153 c#106; j 89 107 6B 153 c#106; j 89 107 6B 153 c#106; j 89 107 6B 153 c#107; k 89 107 6B 153 c#106; j 89 107 6B 153 c#107; k 89 107 6B 153 c#106; j 89 107 6B 153 c#107; k 89 107 6B 153 c#106; j 89 107 6B 153 c#107; k 89 107 6B 153 c#106; j 89 107 6B 153 c#106; j 89 107 6B 153 c#107; k 89 107 6B 153 c#106; j 89 107 6B 153 c#107; k 89 107 6B 153 c#106; j 89 107 6B 153 c#106; j 89 107 6B 153 c#106; j 89 107 6B 153 c#107; k 89 107 6B 153 c#106; j 89 107 6B 153 c#107; k 89 107 6B 153 c#106; j 89 107
10 A 012 LF (NL line feed, new line)
11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 28 053 6#43; + 44 2C 054 6#44; , 45 2D 055 6#45; - 46 2E 056 6#46; . 47 2F 057 6#47; / 48 30 060 6#48; 0 49 31 061 6#49; 1 50 32 062 6#50; 2 51 33 063 6#51; 3 52 34 064 6#52; 4 54 36 066 6#54; 6 55 37 067 6#55; 7 56 38 070 6#56; 8 58 58 130 6#88; X 50 120 6#80; P 50 110 6B 153 6#107; k 76 4C 114 6#76; L 108 6C 154 6#108; 1 77 4D 115 6#77; M 109 6D 155 6#109; m 78 4E 116 6#78; N 110 6E 156 6#110; n 79 4F 117 6#79; 0 111 6F 157 6#111; 0 80 50 120 6#80; P 112 70 160 6#112; P 81 51 121 6#81; Q 81 51 121 6#81; Q 81 53 123 6#83; S 81 55 125 6#85; U 81 77 167 6#119; W 82 58 130 6#88; X 83 58 130 6#88; X 84 54 124 6#86; X 85 57 127 6#87; W 86 56 126 6#86; V 87 57 127 6#87; W 88 58 130 6#88; X
12 C 014 FF (NP form feed, new page) 44 2C 054 , ,
13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 45 2D 055 - - 46 2E 056 . . 47 2F 057 / / 48 30 060 0 0 48 30 060 0 0 48 30 060 0 0 48 30 060 0 0 49 31 061 1 1 50 32 062 2 2 82 52 122 R R 114 72 162 r r 51 33 063 3 3 83 53 123 S S 115 73 163 s s 52 34 064 4 4 84 54 124 T T 116 74 164 t t 53 35 065 5 5 85 55 125 U U 117 75 165 u u 54 36 066 6 6 86 56 126 V V 118 76 166 v v 55 37 067 7 7 87 57 127 W W 119 77 167 w w 56 38 070 8 8 88 58 130 X X 120 78 170 x X
14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 25 38 070 @#56; 8 46 2E 056 @#46; . 47 2F 057 @#47; / 48 30 060 @#48; 0 48 30 060 @#48; 0 48 30 060 @#48; 0 48 30 060 @#48; 0 48 30 060 @#48; 0 49 31 061 @#49; 1 80 50 120 @#80; P 112 70 160 @#112; P 81 51 121 @#81; Q 113 71 161 @#113; Q 114 72 162 @#114; r 115 73 163 @#115; s 115 73 163 @#115; s 115 73 163 @#115; s 115 73 163 @#116; t 116 74 164 @#116; t 117 75 165 @#117; u 118 76 166 @#118; v 119 77 167 @#119; w
15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 47 2F 057 «#47; / 48 30 060 «#48; 0 48 30 060 «#48; 0 49 31 061 «#49; 1 50 32 062 «#50; 2 80 50 120 «#80; P 112 70 160 «#112; P 81 51 121 «#81; Q 113 71 161 «#113; Q 82 52 122 «#82; R 83 53 123 «#83; S 84 54 124 «#84; T 85 55 125 «#85; U 86 56 126 «#86; V 87 57 127 «#87; W 88 58 130 «#88; X 89 50 120 «#80; P 80 50 120 «#80; P 81 51 121 «#81; Q 82 52 122 «#82; R 83 53 123 «#83; S 84 54 124 «#84; T 85 55 125 «#85; U 86 56 126 «#86; V 87 57 127 «#87; W 88 58 130 «#88; X 88 58 130 «#88; X 88 58 130 «#88; X 89 50 120 «#80; P 80 50 120 «#80; P 81 51 121 «#81; Q 81 51 121 «#81; Q 82 52 122 «#82; R 83 53 123 «#83; S 84 54 124 «#84; T 85 55 125 «#85; U 86 56 126 «#86; V 87 57 127 «#87; W 88 58 130 «#88; X 89 50 120 «#80; P 80 50 120 «#80; P 80 50 120 «#80; P 81 51 121 «#12; P 81 51 121 «#81; Q 82 52 122 «#82; R 83 53 123 «#83; S 84 54 124 «#84; T 85 55 125 «#85; U 86 56 126 «#86; V 87 57 127 «#87; W 88 58 130 «#88; X 89 50 120 «#80; P 80 50 120 «#80; P 80 50 120 «#112; P 81 51 121 «#12; P 82 52 122 «#82; R 83 53 123 «#83; S 84 54 124 «#84; T 85 55 125 «#85; U 87 57 127 «#87; W 88 58 130 «#88; X 89 58 130 «#88; X 89 58 130 «#88; X 89 58 130 «#88; X
16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 48 30 060 «#48; 0 49 31 061 «#49; 1 50 32 062 «#50; 2 82 52 122 «#82; R 114 72 162 «#114; r 51 33 063 «#51; 3 83 53 123 «#83; S 115 73 163 «#115; s 84 54 124 «#84; T 116 74 164 «#116; t 85 55 125 «#85; U 117 75 165 «#117; u 86 56 126 «#86; V 87 57 127 «#87; W 88 58 130 «#88; X 88 58 130 «#88; X 89 50 120 «#80; P 80 50 120 «#80; P 81 51 121 «#81; Q 82 52 122 «#82; R 83 53 123 «#83; S 84 54 124 «#84; T 85 55 125 «#85; U 86 56 126 «#86; V 87 57 127 «#87; W 88 58 130 «#88; X 88 58 130 «#88; X 88 58 130 «#88; X 89 50 120 «#80; P 80 50 120 «#80; P 81 51 121 «#81; Q 81 51 121 «#81;
17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 49 31 061 1 1 50 32 062 2 2 50 32 062 2 2 51 33 063 3 3 52 34 064 4 4 53 35 065 5 5 54 36 066 5 5 55 37 067 7 7 65 38 070 7 7 65 38 070 8 8 81 51 121 Q Q 113 71 161 q Q 114 72 162 r r 116 74 164 t t 117 75 165 u u 118 76 166 v v 118 77 167 w w 119 77 167 w w 119 77 167 w w
18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 50 32 062 2 2 51 33 063 3 3 52 34 064 4 4 53 35 065 5 5 54 36 066 5 5 55 37 067 7 7 65 38 070 8 8 50 32 062 2 2 51 33 063 Q 3 52 34 064 4 4 53 35 065 A U 64 54 124 T T 65 67 126 V U 67 116 74 164 t U 68 56 126 V U 68 56 126 V U 68 57 127 W U 78 119 77 167 w U 78 18 030 CAN (cancel) 50 32 062 2 2 51 33 063 3 3 52 34 064 4 4 53 35 065 5 5 65 37 067 7 7 67 57 127 W U 68 58 130 X X
19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 51 33 063 3 3 52 34 064 4 4 53 35 065 5 5 65 34 064 4 4 55 37 067 7 7 66 38 070 8 8 58 53 123 S S 115 73 163 s S 88 58 120 T T 116 74 164 t t 88 58 126 V V 118 76 166 v V 119 77 167 w W 119 77 167 w W
20 14 024 DC4 (device control 4) 52 34 064 4 4 84 54 124 T T 116 74 164 t t 21 15 025 NAK (negative acknowledge) 53 35 065 5 5 85 55 125 U U 117 75 165 u u 22 16 026 SYN (synchronous idle) 54 36 066 6 6 86 56 126 V V 118 76 166 v V 23 17 027 ETB (end of trans. block) 55 37 067 7 7 87 57 127 W W 119 77 167 w W 24 18 030 CAN (cancel) 56 38 070 8 8 88 58 130 X X 120 78 170 x X
21 15 025 NAK (negative acknowledge)
22 16 026 SYN (synchronous idle) 54 36 066 6 6 86 56 126 V V 118 76 166 v V 23 17 027 ETB (end of trans. block) 55 37 067 7 7 87 57 127 W W 119 77 167 w W 24 18 030 CAN (cancel) 56 38 070 8 8 88 58 130 X X 120 78 170 x X
23 17 027 ETB (end of trans. block)
24 18 030 CAN (cancel) 56 38 070 & #56; 8 88 58 130 & #88; X 120 78 170 & #120; X
25 19 031 EM (end of medium) 57 39 071 9 9 89 59 131 Y Y 121 79 171 y Y
26 1A 032 SUB (substitute) 58 3A 072 6#58; : 90 5A 132 6#90; Z 122 7A 172 6#122; Z
27 1B 033 ESC (escape) 59 3B 073 ; ; 91 5B 133 [[123 7B 173 { {
28 1C 034 FS (file separator) 60 3C 074 < < 92 5C 134 \ \ 124 7C 174
29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } }
30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~
31 1F 037 US (unit separator) 63 3F 077 ? ? 95 5F 137 _ _ 127 7F 177 DEL

Source: www.LookupTables.com

StringCompare.java

```
String first = "a";
String second = "A";
int asciia = (int) first.charAt(0);
int asciib = (int) second.charAt(0);
System.out.println("ASCII Code for "+first+" is " + asciia);
System.out.println("ASCII Code for "+second+" is " + asciib);
if (first.compareTo(second) == 0) {
  System.out.println(first+" is equal to "+second);
else if (first.compareTo(second) < 0) {
  System.out.println(first+" is less than "+second);
else if (first.compareTo(second) > 0) {
  System.out.println(first+" is greater than "+second);
```

\$ java StringCompare ASCII Code for a is 97 ASCII Code for A is 65 a is greater than A

Exercise: IsPrimary

Write a program that asks the user for a color and prints whether the color is primary or not.

• The primary colors are "red", "green", "blue"

All other inputs are non-primary

\$ java IsPrimary

Enter a color: **green** green is not primary

\$ java IsPrimary

Enter a color: **blue** blue is primary

Top down design

- 1. Identify features of the program
 - 1. List them out!
- 2. Identify verbs and nouns in feature list
 - 1. Verbs: functions
 - 2. Nouns: objects/variables
- 3. Sketch major steps how features should fit together
 - 1. Algorithm!
- 4. Write program skeleton
 - 1. Include method **stubs** (placeholders for our functions)
 - 2. method **stub**: empty function with parameters and return type
- 5. Implement and test method stubs one at a time

Smart way to wash dishes

Punt the problem to someone else

But we want to wash one dish so we can say we washed a dish

Recursion

a function that calls itself



"Simple" way to solve "similar" problems

Creating a recursive algorithms

Rule that "does work" then "calls itself" on a smaller version of the problem

Base case that handles the smallest problem Prevents "infinite recursion"

Recursion Example – Contains letter

Write a method called "containsLetter" that determines if a String contains a given character

Question: What are the parameters?

- 1. The String to be looking in
- 2. The character to look for

Question: What is the return type?

Recursion Example – Contains letter

How can we break this problem down into smaller problems?

```
contains("l", "apple") =
    contains("l", "a") OR
    contains("l", "p") OR
    contains("l", "p") OR
    contains("l", "l") OR
    contains("l", "e") OR
```

Recursion Visualization — Contains letter

Recursion Example – IndexOf letter

Write a method called IndexOf.

Arguments: String (haystack), Character (needle)

Return: the index of the character in the String, if the chatacter isnt there, return:

-1.

Recursion Example – printVowels

Write a recursive function that prints just the vowels in a String

Recursion limitations

- Limited number of times we can recurse
 - Stackoverflow too many frames
- Potentially memory inefficient
 - If we copy data in subproblems we'll worry about this in a few weeks
- Performance: might duplicate unnecessary work
 - We'll define performance later in the semester

Style

- How we format our programs is very important
 - Like rules of etiquette around eating and keep a clean appearance
 - Like punctuation rules, it helps make text more readable
- Variable names should be descriptive

- Indentation is very important
 - Every statement inside a pair of braces must be indented
- Braces should be placed consistently

Strings revisited

Idea: Store multiple values into a single variable

Values are sequential

Analogous to a list

val

double val = 3.0;

3.0

double[] vals = $\{3.0, 6.0, 7.0, -2.5\}$;

vals

3.0 6.0 7.0 -2.5

Three ways to initialize an array

- 1. With an initial value
 int[] numbers = {1, 2, 5};
- 2. With allocated space, but uninitialized
 int[] numbers = new int[3];
- 3. With an empty array reference
 int[] numbers = null;

Array Indexing

Access individual elements of an array with indexing

Variable name Integer

We use zero-based indexing

first element is 0

last element is length-1

Accessing indices out of range results in a runtime error!

Exercise: print backwards

Write a program, Backwards.java, that asks the user for 3 integers and then prints the list of numbers in reverse order

Strings

Strings are implemented as arrays of characters

```
Get the length of a string with length()

String greeting = "hola";

int len = greeting.length(); // what is the length?

char c = greeting[2]; // what character is in index 2?
```

char: New built-in type, denoted with single quote, e.g. 'a' or '{'

Strings as an array of characters

String str = "hello world"

How many characters in this String?
 10

How do we access the first character?
 str.charAt(0)

How do access the 5th character?
 str.charAt(4)

Exercise: GetCharacters.java

Write a program, GetCharacters.java, that asks the user for a word and then prints the first, last and middle character.

Enter a word: hola!

FirstIndex: 0 FirstCharacter: h

MiddleIndex: 2 MiddleCharacter: I

LastIndex: 5 LastCharacter: !

Command line arguments

```
public static void main(String[] args)
```

Command line arguments are an array of String

Exercise: Write a program called commandLineArgs.java that

- 1) prints out 3 command line arguments that are passed in.
- 2) Compute the sum of three command line arguments (assuming they are integers)

Recursion Example – printList

Write a recursive function that prints the contents of an array