Lecture 8
The % operator

Dividend

\[ 25 \div 7 = 3 \text{ remainder } 4 \]

Divisor

\[ 25 \% 7 = 4 \]

Modulo Operator: returns the remainder of a division
int a = 10, b = 3;
double x = 10.0, y = 3.0;

System.out.println(a % b);
System.out.println(b % a);
System.out.println(x % y);
System.out.println(x % b);
System.out.println(-a % b);
System.out.println(a % -b);

Remainder has same sign with dividend.
Character Data Type and Operations

- The `char` data type.

- is a built-in (primitive) data type of Java
- is used to represent alpha-numerical information (characters) inside the computer
- uses the Unicode to encode characters from many different kinds of languages in the world
- uses 2 bytes of memory to store the Unicode value
- The Unicode includes the ASCII code which is used to encode English characters
Unicode

- Official website for Unicode: [click here]
- Here is a Unicode chart: [click here]

- Unicode provides a unique number for every character

Example:
- The number 65 of Unicode represents the (English) character A
- The number 35 of Unicode represents the character #
- The number 948 of Unicode represents the (Greek) character δ
- The number 24373 of Unicode represents the (Chinese) character 張
ASCII

- The value **0 --- 127** of the **Unicode** is called the **ASCII code**

- The **American Standard Code for Information Interchange (ASCII)** code was designed to *represent* characters in the **English alphabet**

- **ASCII code**: (with letters and number highlighted)
- A **character literal** is written between quotes '...'

- **Examples:**
  - 'A' is the **character literal** A (Unicode code value 65)
  - 'a' is the **character literal** a (Unicode code value 97)
  - '1' is the **character literal** 1 (Unicode code value 49)

```java
public class Char01B {
    public static void main(String[] args) {
        char a;
        a = 65; // 65 = Char code for the character 'A'
        System.out.println(a); // Prints 'A' !!!

        int i;
        i = 'A' + 1; // 'A' is just an integer !!!
        System.out.println(i); // Prints 66 !!!
    }
}
```
Summary of char data type

- A `char` type variable consists of 2 bytes
- It contains the **Unicode code** of some character
- A `char` type variable is an **integer type variable** that contains a **positive number**

A character code is a **(positive) number**

- So: a `char` data type is an integer data type !!!
- The `size` of the `char` data type is 2 bytes
- The `range` of values of the `char` data type is: [0 .. 65535]
public class Char01A
{
    public static void main(String[] args)
    {
        char a, b;
        int i;

        a = 'x';
i = a;       // Safe conversion
System.out.println(i); // Prints 120 (ASCII code for 'x')

        a = 'x';
i = a+1;
System.out.println(i); // Prints 121

        a = 'x';
b = 'y';
i = a + b;
System.out.println(i); // Prints 241 (ASCII code for 'y' = 121)
    }
}
Convert to char

- The range of `byte`, `short`, `int`, and `long` include some negative values.

  The range of values of `char` only contain positive values (including 0).

  Therefore:

  - It is always unsafe to convert any integer type (`byte`, `short`, `int`, and `long`) to `char`.

Recall the following list of unsafe conversions that was discussed previously:

**Unsafe conversions:**

```
double → float → long → int → short → byte
```

Therefore:

- We must use casting (`char`) to assign any kind of integer value to a `char` typed variable.
```java
public class Char03
{
    public static void main(String[] args)
    {
        int x;
        char a;

        x = 98; // 98 is the code for 'b'
        // a = x; // This is not allowed without casting

        a = (char) x; // Convert int to char

        System.out.println(a); // Prints 'b'
    }
}
```
The variable $x$ is assigned the value 98 (which is the Unicode for the character 'b'):

**Program:**

$x = 98; \quad a = \text{(char)} \ x;$

**Variables**

$x: \quad \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
\end{array}$

$a: \quad \text{(char) (2 bytes)}$

(int (4 bytes))

(The bit pattern 00000000 00000000 00000000 01100010 encodes 98 using the binary number system.)
The statement \( a = \text{(char)}\ x; \) will achieve the following:

**Program:**

\[
x = 98; \\
a = \text{(char)}\ x;
\]

**Variables**

- **x:** 00000000 00000000 00000000 01100010
- **a:** 00000000 01100010

```
shortening conversion! Need explicit casting!!
```

This statement **copies the value** from an int typed variable into a char typed variable.
• When a floating-point value is cast into a `char`, the floating-point value is first cast into an `int`, which is then cast into a `char`.

```java
char ch = (char)65.25;  // Decimal 65 is assigned to ch
System.out.println(ch); // ch is character A
```
Consider the **ASCII code table**:

<table>
<thead>
<tr>
<th>0</th>
<th>NUL</th>
<th>1</th>
<th>SOH</th>
<th>2</th>
<th>STX</th>
<th>3</th>
<th>ETX</th>
<th>4</th>
<th>EOT</th>
<th>5</th>
<th>ENQ</th>
<th>6</th>
<th>ACK</th>
<th>7</th>
<th>BEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>BS</td>
<td>9</td>
<td>HT</td>
<td>10</td>
<td>NL</td>
<td>11</td>
<td>VT</td>
<td>12</td>
<td>NP</td>
<td>13</td>
<td>CR</td>
<td>14</td>
<td>SO</td>
<td>15</td>
<td>SI</td>
</tr>
<tr>
<td>16</td>
<td>DLE</td>
<td>17</td>
<td>DC1</td>
<td>18</td>
<td>DC2</td>
<td>19</td>
<td>DC3</td>
<td>20</td>
<td>DC4</td>
<td>21</td>
<td>NAK</td>
<td>22</td>
<td>SYN</td>
<td>23</td>
<td>ETB</td>
</tr>
<tr>
<td>24</td>
<td>CAN</td>
<td>25</td>
<td>EM</td>
<td>26</td>
<td>SUB</td>
<td>27</td>
<td>ESC</td>
<td>28</td>
<td>FS</td>
<td>29</td>
<td>GS</td>
<td>30</td>
<td>RS</td>
<td>31</td>
<td>US</td>
</tr>
<tr>
<td>32</td>
<td>SP</td>
<td>33</td>
<td>!</td>
<td>34</td>
<td>&quot;</td>
<td>35</td>
<td>#</td>
<td>36</td>
<td>$</td>
<td>37</td>
<td>%</td>
<td>38</td>
<td>&amp;</td>
<td>39</td>
<td>'</td>
</tr>
<tr>
<td>40</td>
<td>(</td>
<td>41</td>
<td>)</td>
<td>42</td>
<td>*</td>
<td>43</td>
<td>+</td>
<td>44</td>
<td>,</td>
<td>45</td>
<td>-</td>
<td>46</td>
<td>.</td>
<td>47</td>
<td>/</td>
</tr>
<tr>
<td>48</td>
<td>0</td>
<td>49</td>
<td>1</td>
<td>50</td>
<td>2</td>
<td>51</td>
<td>3</td>
<td>52</td>
<td>4</td>
<td>53</td>
<td>5</td>
<td>54</td>
<td>6</td>
<td>55</td>
<td>7</td>
</tr>
<tr>
<td>56</td>
<td>8</td>
<td>57</td>
<td>9</td>
<td>58</td>
<td>:</td>
<td>59</td>
<td>;</td>
<td>60</td>
<td>&lt;</td>
<td>61</td>
<td>=</td>
<td>62</td>
<td>&gt;</td>
<td>63</td>
<td>?</td>
</tr>
<tr>
<td>64</td>
<td>@</td>
<td>65</td>
<td>A</td>
<td>66</td>
<td>B</td>
<td>67</td>
<td>C</td>
<td>68</td>
<td>D</td>
<td>69</td>
<td>E</td>
<td>70</td>
<td>F</td>
<td>71</td>
<td>G</td>
</tr>
<tr>
<td>72</td>
<td>H</td>
<td>73</td>
<td>I</td>
<td>74</td>
<td>J</td>
<td>75</td>
<td>K</td>
<td>76</td>
<td>L</td>
<td>77</td>
<td>M</td>
<td>78</td>
<td>N</td>
<td>79</td>
<td>O</td>
</tr>
<tr>
<td>80</td>
<td>P</td>
<td>81</td>
<td>Q</td>
<td>82</td>
<td>R</td>
<td>83</td>
<td>S</td>
<td>84</td>
<td>T</td>
<td>85</td>
<td>U</td>
<td>86</td>
<td>V</td>
<td>87</td>
<td>W</td>
</tr>
<tr>
<td>88</td>
<td>X</td>
<td>89</td>
<td>Y</td>
<td>90</td>
<td>Z</td>
<td>91</td>
<td>[</td>
<td>92</td>
<td>\</td>
<td>93</td>
<td>]</td>
<td>94</td>
<td>^</td>
<td>95</td>
<td>_</td>
</tr>
<tr>
<td>96</td>
<td>`</td>
<td>97</td>
<td>a</td>
<td>98</td>
<td>b</td>
<td>99</td>
<td>c</td>
<td>100</td>
<td>d</td>
<td>101</td>
<td>e</td>
<td>102</td>
<td>f</td>
<td>103</td>
<td>g</td>
</tr>
<tr>
<td>104</td>
<td>h</td>
<td>105</td>
<td>i</td>
<td>106</td>
<td>j</td>
<td>107</td>
<td>k</td>
<td>108</td>
<td>l</td>
<td>109</td>
<td>m</td>
<td>110</td>
<td>n</td>
<td>111</td>
<td>o</td>
</tr>
<tr>
<td>112</td>
<td>p</td>
<td>113</td>
<td>q</td>
<td>114</td>
<td>r</td>
<td>115</td>
<td>s</td>
<td>116</td>
<td>t</td>
<td>117</td>
<td>u</td>
<td>118</td>
<td>v</td>
<td>119</td>
<td>w</td>
</tr>
<tr>
<td>120</td>
<td>x</td>
<td>121</td>
<td>y</td>
<td>122</td>
<td>z</td>
<td>123</td>
<td>{</td>
<td>124</td>
<td></td>
<td></td>
<td>125</td>
<td>}</td>
<td>126</td>
<td>~</td>
<td>127</td>
</tr>
</tbody>
</table>

**How to** detect a **lower case letter**:

- When the **ASCII code** of the **character** is
  - $\geq 'a' \text{ (or 97)}$, and
  - $\leq 'z' \text{ (or 122)}$
Consider the ASCII code table:

<p>| | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NUL</td>
<td>1</td>
<td>SOH</td>
<td>2</td>
<td>STX</td>
<td>3</td>
<td>ETX</td>
<td>4</td>
<td>EOT</td>
<td>5</td>
<td>ENQ</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>BS</td>
<td>9</td>
<td>HT</td>
<td>10</td>
<td>NL</td>
<td>11</td>
<td>VT</td>
<td>12</td>
<td>NP</td>
<td>13</td>
<td>CR</td>
<td>14</td>
</tr>
<tr>
<td>16</td>
<td>DLE</td>
<td>17</td>
<td>DC1</td>
<td>18</td>
<td>DC2</td>
<td>19</td>
<td>DC3</td>
<td>20</td>
<td>DC4</td>
<td>21</td>
<td>NAK</td>
<td>22</td>
</tr>
<tr>
<td>24</td>
<td>CAN</td>
<td>25</td>
<td>EM</td>
<td>26</td>
<td>SUB</td>
<td>27</td>
<td>ESC</td>
<td>28</td>
<td>FS</td>
<td>29</td>
<td>GS</td>
<td>30</td>
</tr>
<tr>
<td>32</td>
<td>SP</td>
<td>33</td>
<td>!</td>
<td>34</td>
<td>&quot;</td>
<td>35</td>
<td>#</td>
<td>36</td>
<td>$</td>
<td>37</td>
<td>%</td>
<td>38</td>
</tr>
<tr>
<td>40</td>
<td>(</td>
<td>41</td>
<td>)</td>
<td>42</td>
<td>*</td>
<td>43</td>
<td>+</td>
<td>44</td>
<td>,</td>
<td>45</td>
<td>-</td>
<td>46</td>
</tr>
<tr>
<td>48</td>
<td>0</td>
<td>49</td>
<td>1</td>
<td>50</td>
<td>2</td>
<td>51</td>
<td>3</td>
<td>52</td>
<td>4</td>
<td>53</td>
<td>5</td>
<td>54</td>
</tr>
<tr>
<td>56</td>
<td>8</td>
<td>57</td>
<td>9</td>
<td>58</td>
<td>:</td>
<td>59</td>
<td>;</td>
<td>60</td>
<td>&lt;</td>
<td>61</td>
<td>=</td>
<td>62</td>
</tr>
<tr>
<td>64</td>
<td>@</td>
<td>65</td>
<td>A</td>
<td>66</td>
<td>B</td>
<td>67</td>
<td>C</td>
<td>68</td>
<td>D</td>
<td>69</td>
<td>E</td>
<td>70</td>
</tr>
<tr>
<td>72</td>
<td>H</td>
<td>73</td>
<td>I</td>
<td>74</td>
<td>J</td>
<td>75</td>
<td>K</td>
<td>76</td>
<td>L</td>
<td>77</td>
<td>M</td>
<td>78</td>
</tr>
<tr>
<td>80</td>
<td>P</td>
<td>81</td>
<td>Q</td>
<td>82</td>
<td>R</td>
<td>83</td>
<td>S</td>
<td>84</td>
<td>T</td>
<td>85</td>
<td>U</td>
<td>86</td>
</tr>
<tr>
<td>88</td>
<td>X</td>
<td>89</td>
<td>Y</td>
<td>90</td>
<td>Z</td>
<td>91</td>
<td>[</td>
<td>92</td>
<td>\</td>
<td>93</td>
<td>]</td>
<td>94</td>
</tr>
<tr>
<td>96</td>
<td>`</td>
<td>97</td>
<td>a</td>
<td>98</td>
<td>b</td>
<td>99</td>
<td>c</td>
<td>100</td>
<td>d</td>
<td>101</td>
<td>e</td>
<td>102</td>
</tr>
<tr>
<td>104</td>
<td>h</td>
<td>105</td>
<td>i</td>
<td>106</td>
<td>j</td>
<td>107</td>
<td>k</td>
<td>108</td>
<td>l</td>
<td>109</td>
<td>m</td>
<td>110</td>
</tr>
<tr>
<td>112</td>
<td>p</td>
<td>113</td>
<td>q</td>
<td>114</td>
<td>r</td>
<td>115</td>
<td>s</td>
<td>116</td>
<td>t</td>
<td>117</td>
<td>u</td>
<td>118</td>
</tr>
<tr>
<td>120</td>
<td>x</td>
<td>121</td>
<td>y</td>
<td>122</td>
<td>z</td>
<td>123</td>
<td>{</td>
<td>124</td>
<td></td>
<td></td>
<td>125</td>
<td>}</td>
</tr>
</tbody>
</table>

How to detect an upper case letter:

- When the ASCII code of the character is
  - ≥ 'A' (or 65), and
  - ≤ 'Z' (or 90)
We can convert an Unicode code for a lower case letter into the same upper case letter by subtracting 32 from the code value.

```java
public class Char05 {
    public static void main(String[] args) {
        char x;
        char y;

        x = 'a';
        y = (char) (x - 32);  // We need to cast, because (x - 32) is an int value!
        System.out.println("x = " + x + ", y = " + y + ");

        x = 'b';
        y = (char) (x - 32);
        System.out.println("x = " + x + ", y = " + y + ");

        x = 'c';
        y = (char) (x - 32);
        System.out.println("x = " + x + ", y = " + y + ");
    }
}
```
public class Char05a
{
    public static void main(String[] args)
    {
        char x;
        char y;

        x = 'a';
        y = (char) (x - 'a' + 'A');
        System.out.println("x = " + x + ", y = " + y + ");

        x = 'b';
        y = (char) (x - 'a' + 'A');
        System.out.println("x = " + x + ", y = " + y + ");

        x = 'c';
        y = (char) (x - 'a' + 'A');
        System.out.println("x = " + x + ", y = " + y + ");
    }
}
Class Echo360

- [Link](https://playback.service.emory.edu/ess/login.jsp)
• QUIZ!