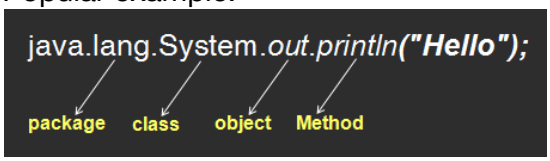
# Week3 Notes

### Primitive Data Types

- Math library of Java: You can use for more complex mathematical operations such as sine, cosine, logarithm etc.
- For now, you can memorize some use cases we discussed but it's good to know why/how we use the periods in that syntax.
- Popular example:




Source: Smartherd

- Function (Method): A block of code that performs a specific task. There are several built-in functions, e.g., math.sqrt(). You can design your own functions hopefully in 3 weeks 🙂
- You can increment/decrement chars similar to an integer. You can even initialize them as integers.
- Every char has an ASCII number (they are basically integers, see figure given below). You access any char with the associated numeric value.

## ASCII Table

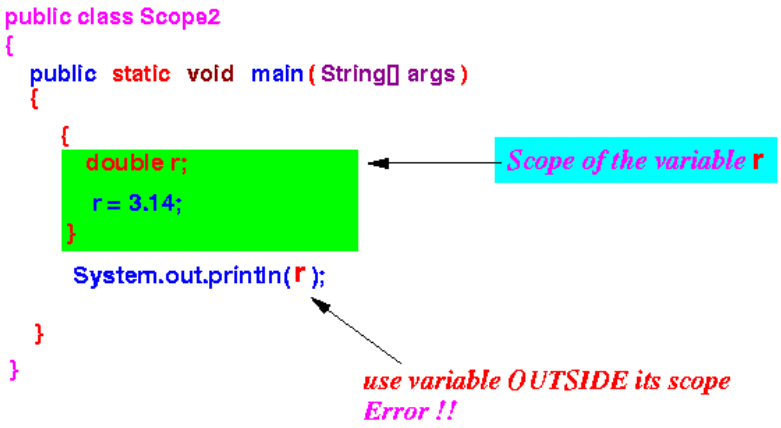
| Dec | Hex | Oct | Char | Dec | Hex | Oct | Char | Dec | Hex | Oct | Char | Dec | Hex | Oct | Char |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | | 32 | 20 | 40 | [space] | 64 | 40 | 100 | @ | 96 | 60 | 140 | ` |
| 1 | 1 | 1 | | 33 | 21 | 41 | ! | 65 | 41 | 101 | A | 97 | 61 | 141 | a |
| 2 | 2 | 2 | | 34 | 22 | 42 | " | 66 | 42 | 102 | B | 98 | 62 | 142 | b |
| 3 | 3 | 3 | | 35 | 23 | 43 | # | 67 | 43 | 103 | C | 99 | 63 | 143 | c |
| 4 | 4 | 4 | | 36 | 24 | 44 | $ | 68 | 44 | 104 | D | 100 | 64 | 144 | d |
| 5 | 5 | 5 | | 37 | 25 | 45 | % | 69 | 45 | 105 | E | 101 | 65 | 145 | e |
| 6 | 6 | 6 | | 38 | 26 | 46 | & | 70 | 46 | 106 | F | 102 | 66 | 146 | f |
| 7 | 7 | 7 | | 39 | 27 | 47 | ' | 71 | 47 | 107 | G | 103 | 67 | 147 | g |
| 8 | 8 | 10 | | 40 | 28 | 50 | ( | 72 | 48 | 110 | H | 104 | 68 | 150 | h |
| 9 | 9 | 11 | | 41 | 29 | 51 | ) | 73 | 49 | 111 | I | 105 | 69 | 151 | i |
| 10 | A | 12 | | 42 | 2A | 52 | * | 74 | 4A | 112 | J | 106 | 6A | 152 | j |
| 11 | B | 13 | | 43 | 2B | 53 | + | 75 | 4B | 113 | K | 107 | 6B | 153 | k |
| 12 | C | 14 | | 44 | 2C | 54 | , | 76 | 4C | 114 | L | 108 | 6C | 154 | l |
| 13 | D | 15 | | 45 | 2D | 55 | - | 77 | 4D | 115 | M | 109 | 6D | 155 | m |
| 14 | E | 16 | | 46 | 2E | 56 | . | 78 | 4E | 116 | N | 110 | 6E | 156 | n |
| 15 | F | 17 | | 47 | 2F | 57 | / | 79 | 4F | 117 | O | 111 | 6F | 157 | o |
| 16 | 10 | 20 | | 48 | 30 | 60 | 0 | 80 | 50 | 120 | P | 112 | 70 | 160 | p |
| 17 | 11 | 21 | | 49 | 31 | 61 | 1 | 81 | 51 | 121 | Q | 113 | 71 | 161 | q |
| 18 | 12 | 22 | | 50 | 32 | 62 | 2 | 82 | 52 | 122 | R | 114 | 72 | 162 | r |
| 19 | 13 | 23 | | 51 | 33 | 63 | 3 | 83 | 53 | 123 | S | 115 | 73 | 163 | s |
| 20 | 14 | 24 | | 52 | 34 | 64 | 4 | 84 | 54 | 124 | T | 116 | 74 | 164 | t |
| 21 | 15 | 25 | | 53 | 35 | 65 | 5 | 85 | 55 | 125 | U | 117 | 75 | 165 | u |
| 22 | 16 | 26 | | 54 | 36 | 66 | 6 | 86 | 56 | 126 | V | 118 | 76 | 166 | v |
| 23 | 17 | 27 | | 55 | 37 | 67 | 7 | 87 | 57 | 127 | W | 119 | 77 | 167 | w |
| 24 | 18 | 30 | | 56 | 38 | 70 | 8 | 88 | 58 | 130 | X | 120 | 78 | 170 | x |
| 25 | 19 | 31 | | 57 | 39 | 71 | 9 | 89 | 59 | 131 | Y | 121 | 79 | 171 | y |
| 26 | 1A | 32 | | 58 | 3A | 72 | : | 90 | 5A | 132 | Z | 122 | 7A | 172 | z |
| 27 | 1B | 33 | | 59 | 3B | 73 | ; | 91 | 5B | 133 | [ | 123 | 7B | 173 | { |
| 28 | 1C | 34 | | 60 | 3C | 74 | < | 92 | 5C | 134 | \ | 124 | 7C | 174 | \| |
| 29 | 1D | 35 | | 61 | 3D | 75 | = | 93 | 5D | 135 | ] | 125 | 7D | 175 | } |
| 30 | 1E | 36 | | 62 | 3E | 76 | > | 94 | 5E | 136 | ^ | 126 | 7E | 176 | ~ |
| 31 | 1F | 37 | | 63 | 3F | 77 | ? | 95 | 5F | 137 | _ | 127 | 7F | 177 | |

Source: Harvard CS50 Course (you may want to review this course)

- Long to integer conversion: Since integer has smaller domain, compiler gives error. Same error on integer to char conversion because there are fewer characters than integers.
- Compiler converts any validation statement true or false (e.g. (10>9)) to boolean.

**Scope of Variables**

- A code block is statements written between curly braces.
- Scope is the region where defined variables are usable.
- The scope of a variable depends on where you define a variable. It starts right after the point of declaration, goes until the end of the block.
- A variable is not usable outside (i.e. before/after) its scope.
- In the scope of a variable, you cannot define another variable with the same name.
- You cannot refer to a variable before its declaration.
- Scope example:




Source: Emory University

- Initialization error: When you declare a variable and use it possibly without initialization.
- To be on the safe side, initialize variables when/after you declare.

**Input Mechanisms**

- User input is dynamically changing in real life.
- System.in.read() : getting input from user (once).
- Casting: changing data type into a different data type. Usage: new data type in parenthesis, e.g., (double)
- Some escape characters: "\n" : new line "\t “ : tab
- Using "java classname inputs” command is another option to receive user input.
- Arrays - A quick preview:
    - Consider arrays as vectors

    - Syntax:  datatype[] arrayname (we will cover arrays in detail later)
    - Index (address) of an array starts with 0
    - You can access any element of an array by using its index. Example: arrayname[5] (returns 6th element of array)
- When we have our usual main method line, args array is automatically created for you. It's actually what's input to the main method. We will cover these in detail while discussion methods.
- Double.parseDouble(data): casts data into a double
- Integer.parseInt(data): casts data into an integer
- We have to make sure these types can be casted. Not all casting is possible or makes sense. But a string of say '6' can definitely be converted to an integer.
- Remember some casting is easier as mentioned on the third bullet above; (double) would convert an int to a double.

We have covered some topics from future chapters too but make sure that you read up to Pg.69 in the textbook.