**3.01: Pseudocode (Algorithmic thinking)**

**3.02: Primitive Data Types: Char**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Character Literals** | | 'A' | | 'p' | | '-' | | '8' | | |  | | --- | | **String Literals** | | "Java" | | "Computer Science" | | "555-55-5555" | | "8" | |

* System.out.println('8' + '2');
  + 82
* System.out.println("128" + "32");
  + 12832
* ASCII (0-255):
  + uppercase and lowercase letters
  + punctuation marks
  + the digits 0-9
  + spaces
  + special symbols (e.g. ‘&’)
  + un-printable control (e.g. the Enter key).
* Casting can be used for chars in ASCII
  + System.out.println("The ASCII code for " + 'A' + " is " + (int)'A');
    - The ASCII code for A is 65
  + System.out.println("173 is the ASCII code for " + (char)173);
    - 173 is the ASCII code for –

**3.03: String Objects Lite**

* If an identifier is declared as a **String**, then keyboard characters can be assigned to it
  + String course = "AP Computer Science";  
    String lastName = "Gulliver";  
    String phoneNumber = "555-555-5555";  
    String symbol = "a";
  + Because String is a class, it begins with a capital letter.
* Since **String** is not a primitive data type, it must be a **reference variable**, which is also called an **object.**

**3.04: Escape Sequences**

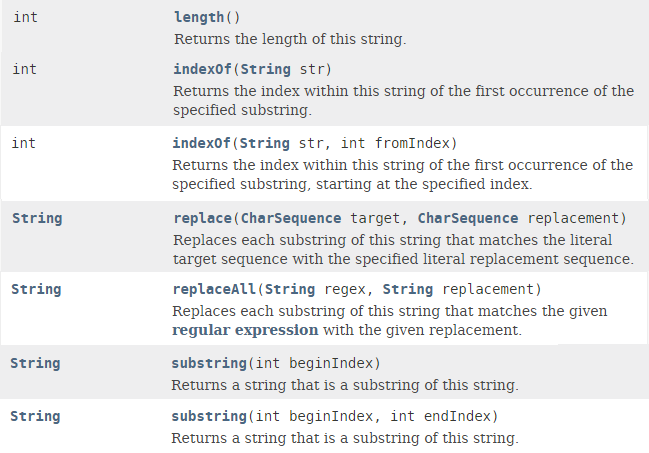
* **Escape sequences** – a pair of symbols beginning with a backslash that has a special meaning within a print statement.

| **Escape Sequences and Their Corresponding Actions** | |
| --- | --- |
| **Escape Sequence** | **Action** |
| \n | new line |
| \t | tab |
| \" | double quote |
| \' | single quote |
| \\ | backslash |

* System.out.println("Not everything in \"quotes\" can be printed!");
  + Output: Not everything in “quotes” can be printed!

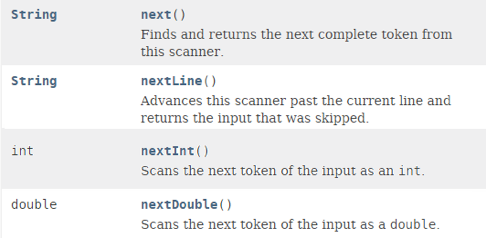
**3.05: Java API**

N.A

**3.06: String Class Method – The Basics**

* The Method Summary table tells you four important pieces of information about each method.
  + The name of the method
  + A description of the method’s purpose
  + How many parameters the method takes and the type of each one
  + The type of information the method returns after it completes its task
* **Dot notation -**  the notion used to call methods or access variables
  + String phrase = "Four score and seven years ago";  
    int stringLength = phrase.length();  
    System.out.println("Length of String: " + stringLength);  
    System.out.println();
* String s="hello";
* System.out.println(s.substring(0,2));//he
  + In the above substring, 0 points to h but 2 points to e (because end index is exclusive).

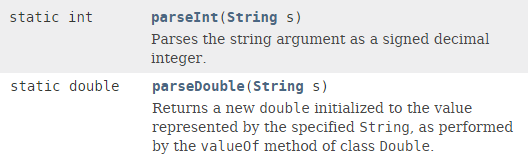
**3.07: Scanner Class Methods**



* These four methods, nextInt(), nextDouble(), next(), and nextLine() can accept simple numeric or alphanumeric input from the keyboard.
* A Scanner class object can be created as follows:
  + Scanner in = new Scanner(System.in);
* With a Scanner object declared, the following statement could be written to accept a decimal number typed in from the keyboard:
  + double number = in.nextDouble();
  + Dot notation is used to separate the object on the left from the method on the right.
* Objects can be declared on one line or two; the choice depends on the context of the program.
  + Defining an object in 2 lines:
    - ClassName objectName;  
      objectName = new ClassName(parameter list);
  + Defining an object in 1 line:
    - ClassName objectName = new ClassName( parameter list);
* next() only takes the first line of an input while nextLine() takes all the lines of an input
  + EX: Inputting “Hello world!” would only take “Hello” and pass “world!” to the next method

**3.08: Parsing**

* **Parse** – to resolve into component parts
  + In computer science, it means to separate into more easily processed parts.
* Java has two methods for removing, or parsing, a number from its String wrapper so it can be used once again in an arithmetic expression to perform calculations.



* Members of the Integer and Double classes respectively