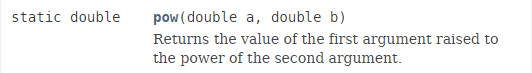
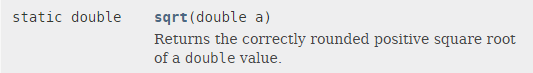
**7.01: JAVA’S MATH CLASS**

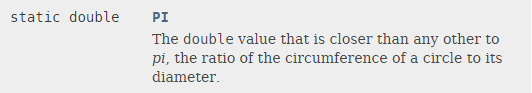
* With Java, to calculate 215 you could write the following statement.
  + EX: int value = 2 \* 2 \* 2 \* 2 \* 2 \* 2 \* 2 \* 2 \* 2 \* 2 \* 2 \* 2 \* 2 \* 2 \* 2;
  + Brute force; not very effective
* **Java’s Math class** has a method for this
  + pow()
  + EX: int value = Math.pow(2,15);
    - Math.pow() invokes the pow() method of the Math class.
    - Two arguments, the values assigned to number and nthPower, are passed to the pow() method.
    - The pow() method computes numbernthpower and assigns the calculated value of 215 to the variable powValue.
* When using the Math class, you must import from java.lang.Math and use Math.METHODNAME





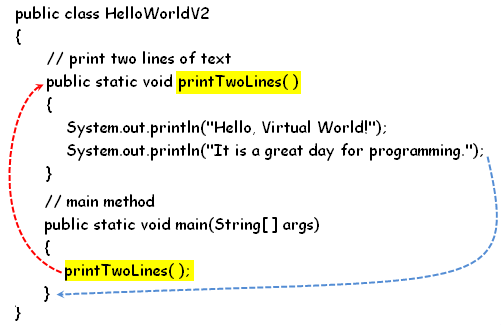






**7.02: DEFINING NEW STATIC METHODS – PART 1**

* **Method** – a segment of code that performs a specific task when invoked by a message statement
  + The main() method is what we have been using up to this point
    - Gets cluttered easily
* **Top-down design** – a style of coding in which programs are broken down into smaller subtasks
  + Breaks a program down into modular components
  + Methods are the workhorses of Java



**7.02: DEFINING NEW STATIC METHODS – PART 2**

* The main() methods primary task is now to manage the flow to control and invoke other methods.