

Goal: explore various algorithms for solving problems using the primitive data types in Java

- (a) Use only the *four primitive types*: char, int, double, boolean, and learn about the Scanner class.
  - (b) Do not use any methods in the Math class.
  - (c) Verify your work with a menu-driven program.
1. Write a method `factors(int num)` that will print all the factors of a given positive integer. For example, `factors(30)` should produce the following formatted output (note that a period terminates the list):  
The factors of 30 are: 1, 2, 3, 5, 6, 10, 15, 30.
  2. Write a method `GCD(int a, int b)` that returns the greatest common divisor of its two positive integer parameters.
  3. Write a boolean method `prime(int num)` that determines whether a given integer greater than one is a prime number. Use this header: `boolean prime(int num)`
  4. Write a method `double power (double base, int exponent)` that raises a given number (real or integer) to a given (positive, negative, or zero) integer power. Do NOT use `pow(x)` or `log(x)` in your solution. Note that if the base is zero, the exponent must be positive.
  5. Write a method `findDigit (int num, int n)` that returns the  $n^{\text{th}}$  digit from the right of a given integer where  $n$  is a positive integer. For example,  
`findDigit (30568,2)` will return 6  
`findDigit(234,5)` will return 0  
`findDigit(-4532,3)` will return 5
  6. Write a method `downDigits(int num)` that will list the digits of a positive integer in one column. For Example:  
  
downDigits (560) will produce the output at right

The digits of 560 are:
5
6
0
  7. Write a method `countDigits (double num)` that returns the number of digits to the left of the decimal point of a valid decimal point number. Note that `countDigits (0.74)` returns 1.