Basic Core Programs

1. **Flip Coin and print percentage of Heads and Tails**
    a. I/P -> The number of times to Flip Coin. *Ensure it is a positive integer*.
    b. Logic -> Use Random Function to get value between 0 and 1. If < 0.5 then tails or heads
    c. O/P -> Percentage of Head vs Tails
2. **Leap Year**
    a. I/P -> Year, ensure it is a 4 digit number.
    b. Logic -> Determine if it is a Leap Year.
    c. O/P -> Print the year is a Leap Year or not.
3. **Power of 2**
    a. Desc -> This program takes a command-line argument N and prints a table of the powers of 2 that are less than or equal to 2^N.
    b. I/P -> The Power Value N. *Only works if 0 <= N < 31 since 2^31 overflows an int*
    c. Logic -> repeat until i equals N.
    d. O/P -> Print the year is a Leap Year or not.
4. **Harmonic Number**
    a. Desc -> Prints the Nth harmonic number: 1/1 + 1/2 + ... + 1/N (http://users.encs.concordia.ca/~chvatal/notes/harmonic.html).
    b. I/P -> The Harmonic Value N. *Ensure N != 0*
    c. Logic -> compute 1/1 + 1/2 + 1/3 + ... + 1/N
    d. O/P -> Print the Nth Harmonic Value.
5. **Factors**
    a. Desc -> Computes the prime factorization of N using brute force.
    b. I/P -> Number to find the prime factors
    c. Logic -> Traverse till i*i <= N instead of i <= N for efficiency.
    d. O/P -> Print the prime factors of number N.

6. **Java Program to Compute Quotient and Remainder**
7. **Java Program to Swap Two Numbers**
8. **Java Program to Check Whether a Number is Even or Odd**
9. **Java Program to Check Whether an Alphabet is Vowel or Consonant**
10. **Java Program to Find the Largest Among Three Numbers**

Functional Programs

1. **2D Array**

    a. Desc -> A library for reading in 2D arrays of integers, doubles, or booleans from standard input and printing them out to standard output.
    b. I/P -> M rows, N Cols, and M * N inputs for 2D Array. Use Java Scanner Class
    c. Logic -> create 2 dimensional array in memory to read in M rows and N cols
    d. O/P -> Print function to print 2 Dimensional Array. In Java use PrintWriter with OutputStreamWriter to print the output to the screen.

2. **Sum of three Integer adds to ZERO**

    a. Desc -> A program with cubic running time. Read in N integers and counts the number of triples that sum to exactly 0.
    b. I/P -> N number of integer, and N integer input array
    c. Logic -> Find distinct triples (i, j, k) such that a[i] + a[j] + a[k] = 0
    d. O/P -> One Output is number of distinct triplets as well as the second output is to print the distinct triplets.

3. Write a program ***Distance.java*** that takes two integer command-line arguments x and y and prints the Euclidean distance from the point (x, y) to the origin (0, 0). The formulae to calculate distance = sqrt(x*x + y*y). Use Math.power function

4. Write a program ***Quadratic.java*** to find the roots of the equation a*x*x + b*x + c. Since the equation is x*x, hence there are 2 roots. The 2 roots of the equation can be found using a formula (Note: Take a, b and c as input values)

    delta = b*b - 4*a*c

    Root 1 of x = (-b + sqrt(delta))/(2*a)

    Root 2 of x = (-b - sqrt(delta))/(2*a)

5. Write a program ***WindChill.java*** that takes two double command-line arguments t and v and prints the wind chill. Use Math.pow(a, b) to compute ab. Given the temperature t (in Fahrenheit) and the wind speed v (in miles per hour), the National Weather Service defines the effective temperature (the wind chill) to be:

    $$w = 35.74 + 0.6215\,t + (0.4275\,t - 35.75)\,v^{0.16}$$

    Note: the formula is not valid if t is larger than 50 in absolute value or if v is larger than 120 or less than 3 (you may assume that the values you get are in that range).