

Functional Programs

1. User Input and Replace String Template “Hello <<UserName>>, How are you?”

- a. I/P -> Take User Name as Input. *Ensure UserName has min 3 char*
- b. Logic -> Replace <<UserName>> with the proper name
- c. O/P -> Print the String with User Name

2. Flip Coin and print percentage of Heads and Tails

- a. I/P -> The number of times to Flip Coin. *Ensure it is 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

3. 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.

4. 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 $0 \leq N < 31$ 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.

5. Harmonic Number

- a. Desc -> Prints the Nth harmonic number: $1/1 + 1/2 + \dots + 1/N$
(<http://users.encs.concordia.ca/~chvatal/notes/harmonic.html>).
- b. I/P -> The Harmonic Value N. *Ensure $N \neq 0$*
- c. Logic -> compute $1/1 + 1/2 + 1/3 + \dots + 1/N$
- d. O/P -> Print the Nth Harmonic Value.

6. 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 $i*i \leq N$ instead of $i \leq N$ for efficiency.
- d. O/P -> Print the prime factors of number N.

7. Gambler

- a. Desc -> Simulates a gambler who start with \$stake and place fair \$1 bets until he/she goes broke (i.e. has no money) or reach \$goal. Keeps track of the number of times he/she wins and the number of bets he/she makes. Run the experiment N times, averages the results, and prints them out.
- b. I/P -> \$Stake, \$Goal and Number of times
- c. Logic -> Play till the gambler is broke or has won
- d. O/P -> Print Number of Wins and Percentage of Win and Loss.

8. Coupon Numbers

- a. Desc -> Given N distinct Coupon Numbers, how many random numbers do you need to generate distinct coupon number? This program simulates this random process.
- b. I/P -> N Distinct Coupon Number
- c. Logic -> repeatedly choose a random number and check whether it's a new one.
- d. O/P -> total random number needed to have all distinct numbers.
- e. Functions => Write Class Static Functions to generate random number and to process distinct coupons.

9. 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 Ncols
- d. O/P -> Print function to print 2 Dimensional Array. In Java use PrintWriter with OutputStreamWriter to print the output to the screen.

10. 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.

11. 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 = $\sqrt{x^2 + y^2}$. Use Math.power function

12. Write static functions to return all permutation of a String using iterative method and Recursion method. Check if the arrays returned by two string functions are equal.

13. Simulate Stopwatch Program

- a. Desc -> Write a Stopwatch Program for measuring the time that elapses between the start and end clicks
- b. I/P -> Start the Stopwatch and End the Stopwatch
- c. Logic -> Measure the elapsed time between start and end
- d. O/P -> Print the elapsed time.

14. Cross Game or Tic-Tac-Toe Game



- a. Desc -> Write a Program to play a Cross Game or Tic-Tac-Toe Game. Player 1 is the Computer and the Player 2 is the user. Player 1 take Random Cell that is the Column and Row.
- b. I/P -> Take User Input for the Cell i.e. Col and Row to Mark the 'X'
- c. Logic -> The User or the Computer can only take the unoccupied cell. The Game is played till either wins or till draw...
- d. O/P -> Print the Col and the Cell after every step.
- e. Hint -> The Hints is provided in the Logic. Use Functions for the Logic...

15. Write a program **Quadratic.java** to find the roots of the equation $a*x^2 + b*x + c$. Since the equation is x^2 , hence there are 2 roots. The 2 roots of the equation can be found using a formula

$$\text{delta} = b*b - 4*a*c$$

$$\text{Root 1 of } x = (-b +$$

$$\text{sqrt(delta))/(2*a) Root 2 of}$$

$$x = (-b - \text{sqrt(delta)})/(2*a)$$

Take a, b and c as input values to find the roots of x.

16. 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 a^b .

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).