**Problem Statements:**

1. **Number Guessing Game**:

○ Write a Java program where the user thinks of a number between 1 and 100, and the computer tries to guess the number by generating random guesses.

○ The user provides feedback by indicating whether the guess is **high**, **low**, or **correct**. The program should be modular, with different functions for generating guesses, receiving user feedback, and determining the next guess.

import java.util.Scanner;

import java.lang.Math;

class NumberGuessGame{

public static void checker(int user, int comp){

Scanner input = new Scanner(System.in);

while(true){

if(user==comp) {

System.out.println("You guessed the correct number");

break;

}

else if(user>comp){

System.out.println("Go Low");

System.out.print("Guess the number : ");

user = input.nextInt();

}

else if(user<comp){

System.out.println("Go High");

System.out.print("Guess the number : ");

user = input.nextInt();

}

}

input.close();

}

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Guess the number : ");

int user = input.nextInt();

int comp = (int)(Math.random()\*100) + 1;

checker(user, comp);

input.close();

}

}

2. **Maximum of Three Numbers**:

○ Write a program that takes three integer inputs from the user and finds the maximum of the three numbers.

○ Ensure your program follows best practices for organizing code into modular functions, such as separate functions for taking input and calculating the maximum value.

import java.util.Scanner;

class MaximumOfThree{

public static int[] takeInput(int size){

Scanner input = new Scanner(System.in);

int[] numbers = new int[size];

for(int i=0; i<size; i++){

System.out.println("Enter number " + (i+1) + " : ");

numbers[i] = input.nextInt();

}

input.close();

return numbers;

}

public static void findMax(int[] numbers){

int max = -1;

for(int i=0; i<numbers.length; i++){

if(numbers[i]>max){

max = numbers[i];

}

}

System.out.println("The maximum number is :" + max);

}

public static void main(String[] args){

int size = 3;

int[] numbers = takeInput(size);

findMax(numbers);

}

}

3. **Prime Number Checker**:

○ Create a program that checks whether a given number is a prime number. ○ The program should use a separate function to perform the prime check and return the result.

import java.util.Scanner;

class PrimeNumberChecker{

public static boolean isPrime(int n){

boolean flag = true;

for(int i=2; i<n; i++){

if(n%i==0) flag = false;

}

return flag;

}

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int n = input.nextInt();

if(isPrime(n)) System.out.println("Its a prime");

else System.out.println("Not a prime");

input.close();

}

}

**Additional Problem Statements:**

4. **Fibonacci Sequence Generator**:

○ Write a program that generates the Fibonacci sequence up to a specified number of terms entered by the user.

○ Organize the code by creating a function that calculates and prints the Fibonacci sequence.

import java.util.Scanner;

class FibonacciGenerator{

public static void generate(int n){

int s1=0;

int s2=1;

while(n>0){

System.out.print(s1 + "");

int s3 = s1+s2;

s1 = s2;

s2 = s3;

n--;

}

}

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int n = input.nextInt();

generate(n);

input.close();

}

}

5. **Palindrome Checker**:

○ Write a program that checks if a given string is a palindrome (a word, phrase, or sequence that reads the same backward as forward).

○ Break the program into functions for input, checking the palindrome condition, and displaying the result.

import java.util.Scanner;

class Palindrome{

public static String takeInput(){

Scanner input = new Scanner(System.in);

String text = input.nextLine();

input.close();

return text;

}

public static boolean checkPalindrome(String text){

String str = "";

for(int i=0; i<text.length(); i++){

char ch = text.charAt(i);

str = ch + str;

}

if(str.equals(text)) return true;

else return false;

}

public static void main(String[] args) {

String text = takeInput();

boolean res = checkPalindrome(text);

if(res) System.out.println("Its a palindrome");

else System.out.println("Not a palindrome");

}

}

6. **Factorial Using Recursion**:

○ Write a program that calculates the factorial of a number using a recursive function.

○ Include modular code to separate input, calculation, and output processes.

import java.util.Scanner;

class Factorial{

public static int takeInput(){

Scanner input = new Scanner(System.in);

int num = input.nextInt();

input.close();

return num;

}

public static int findFactorial(int num){

if(num == 0) return 1;

else {

return num\*findFactorial(num-1);

}

}

public static void main(String[] args) {

int num = takeInput();

int fact = findFactorial(num);

System.out.println("Factorial : " + fact);

}

}

7. **GCD and LCM Calculator**:

○ Create a program that calculates the Greatest Common Divisor (GCD) and Least Common Multiple (LCM) of two numbers using functions.

○ Use separate functions for GCD and LCM calculations, showcasing how modular code works.

import java.util.Scanner;

class GCD\_LCM\_Calculator {

public static int gcd(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

public static int lcm(int a, int b) {

return (a \* b) / gcd(a, b);

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter first number: ");

int num1 = scanner.nextInt();

System.out.print("Enter second number: ");

int num2 = scanner.nextInt();

int gcdResult = gcd(num1, num2);

int lcmResult = lcm(num1, num2);

System.out.println("GCD of " + num1 + " and " + num2 + " is: " + gcdResult);

System.out.println("LCM of " + num1 + " and " + num2 + " is: " + lcmResult);

scanner.close();

}

}

8. **Temperature Converter**:

○ Write a program that converts temperatures between Fahrenheit and Celsius. ○ The program should have separate functions for converting from Fahrenheit to Celsius and from Celsius to Fahrenheit.

import java.util.Scanner;

class TemperatureConverter{

public static double CtoF(double temp){

return (temp\*(9.0/5.0))+32;

}

public static double FtoC(double temp){

return (temp-32)\*(5.0/9.0);

}

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter temperature : ");

double temp = input.nextDouble();

System.out.println("To convert to celsius enter 'c' and 'f' for fahrenheit");

char in = input.next().charAt(0);

if(in=='c'){

double res = FtoC(temp);

System.out.println("Temperature in celsius : " + res);

}

else{

double res = CtoF(temp);

System.out.println("Temperature in Fahrenheit : " + res);

}

input.close();

}

}

9. **Basic Calculator**:

○ Write a program that performs basic mathematical operations (addition, subtraction, multiplication, division) based on user input.

○ Each operation should be performed in its own function, and the program should prompt the user to choose which operation to perform.

import java.util.Scanner;

class BasicCalculator{

public static void operations(int num1, int num2, char ch){

switch(ch){

case '+':

System.out.println("Result : " + (num1+num2));

break;

case '-':

System.out.println("Result : " + (num1-num2));

break;

case '\*':

System.out.println("Result : " + (num1\*num2));

break;

case '/':

System.out.println("Result : " + (num1/num2));

break;

case '%':

System.out.println("Result : " + (num1%num2));

break;

default:

System.out.println("wrong entry");

}

}

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter first number : ");

int num1 = input.nextInt();

System.out.print("Enter second number : ");

int num2 = input.nextInt();

System.out.print("Enter one operation you want to perform( +,-,\*,/,%) : ");

char ch = input.next().charAt(0);

operations(num1, num2, ch);

input.close();

}

}