4. Write a program to add two complex numbers.

**package** org.ruchi;

**import** java.util.Scanner;

**class** ComplexNumber {

**double** real;

**double** imaginary;

// Constructor

ComplexNumber(**double** r, **double** i) {

**this**.real = r;

**this**.imaginary = i;

}

// Method to add two complex numbers

ComplexNumber add(ComplexNumber c) {

**return** **new** ComplexNumber(**this**.real + c.real, **this**.imaginary + c.imaginary);

}

// Method to subtract two complex numbers

ComplexNumber subtract(ComplexNumber c) {

**return** **new** ComplexNumber(**this**.real - c.real, **this**.imaginary - c.imaginary);

}

// Method to display the complex number

**void** display() {

System.***out***.println(**this**.real + " + " + **this**.imaginary + "i");

}

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

// Get user input for the first complex number

System.***out***.print("Enter the real part of the first complex number: ");

**double** real1 = scanner.nextDouble();

System.***out***.print("Enter the imaginary part of the first complex number: ");

**double** imaginary1 = scanner.nextDouble();

ComplexNumber c1 = **new** ComplexNumber(real1, imaginary1);

// Get user input for the second complex number

System.***out***.print("Enter the real part of the second complex number: ");

**double** real2 = scanner.nextDouble();

System.***out***.print("Enter the imaginary part of the second complex number: ");

**double** imaginary2 = scanner.nextDouble();

ComplexNumber c2 = **new** ComplexNumber(real2, imaginary2);

System.***out***.print("First complex number: ");

c1.display();

System.***out***.print("Second complex number: ");

c2.display();

// Adding two complex numbers

ComplexNumber sum = c1.add(c2);

System.***out***.print("Sum: ");

sum.display();

// Subtracting two complex numbers

ComplexNumber difference = c1.subtract(c2);

System.***out***.print("Difference: ");

difference.display();

}

}

9. Write a program to find all the prime numbers from 1 to N.

**package** org.ruchi;

**import** java.util.Scanner;

**public** **class** PrimeNumbers {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the value of N: ");

**int** N = scanner.nextInt();

System.***out***.println("Prime numbers from 1 to " + N + " are:");

**for** (**int** num = 2; num <= N; num++) {

**if** (*isPrime*(num)) {

System.***out***.print(num + " ");

}}}

**public** **static** **boolean** isPrime(**int** num) {

**if** (num <= 1) {

**return** **false**;

}

**for** (**int** i = 2; i <= Math.*sqrt*(num); i++) {

**if** (num % i == 0) {

**return** **false**;

}

}

**return** **true**;

}}

13. Write a program to calculate compound interest.

**package** org.ruchi;

**import** java.util.Scanner;

**public** **class** CompoundInterest {

**public** **static** **void** main(String[] args) {

**float** principal;

**float** rate;

**float** time;

**float** compoundingPeriods;

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the principal amount: ");

principal = scanner.nextFloat();

System.***out***.print("Enter the annual interest rate (in %): ");

rate = scanner.nextFloat();

System.***out***.print("Enter the time (in years): ");

time = scanner.nextFloat();

System.***out***.print("Enter the number of compounding periods per year: ");

compoundingPeriods = scanner.nextFloat();

**double** amount = principal \* Math.*pow*(1 + (rate / (compoundingPeriods \* 100)), compoundingPeriods \* time);

**double** compoundInterest = amount - principal;

System.***out***.println("The compound interest is: " + compoundInterest);

System.***out***.println("The total amount after " + time + " years is: " + amount);

}

}

15. Write a program that prompts the user to input an integer and then outputs the number with the

digits reversed.

**package** org.ruchi;

**import** java.util.Scanner;

**public** **class** ReverseDigits {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter an integer: ");

**int** number = scanner.nextInt();

**int** reversedNumber = 0;

**while** (number != 0) {

**int** digit = number % 10;

reversedNumber = reversedNumber \* 10 + digit;

number = number / 10;

}

System.***out***.println("Reversed number: " + reversedNumber);

}}

16. Write a program to accept two numbers and find the power of each (Do not use Java built-in

method)

**package** org.ruchi;

**import** java.util.Scanner;

**public** **class** PowerCalculator {

**public** **static** **long** power(**int** base, **int** exponent) {

**long** result = 1;

**for** (**int** i = 0; i < exponent; i++) {

result \*= base;

}

**return** result;

}

**public** **static** **void** main(String[] args) {

Scanner s = **new** Scanner(System.***in***);

System.***out***.print("Enter the base number: ");

**int** base = s.nextInt();

System.***out***.print("Enter the exponent: ");

**int** exponent = s.nextInt();

**long** powerResult = *power*(base, exponent);

System.***out***.println(base + " raised to the power of " + exponent + " is: " + powerResult);

}

}

17. Write a program to check Armstrong number between two integers.

**package** org.ruchi;

**import** java.util.Scanner;

**public** **class** Armstrong {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the lower bound: ");

**int** lowerBound = scanner.nextInt();

System.***out***.print("Enter the upper bound: ");

**int** upperBound = scanner.nextInt();

System.***out***.println("Armstrong numbers between " + lowerBound + " and " + upperBound + " are:");

**for** (**int** num = lowerBound; num <= upperBound; num++) {

**if** (*isArmstrong*(num)) {

System.***out***.print(num + " ");

}}}

**public** **static** **boolean** isArmstrong(**int** number) {

**int** originalNumber = number;

**int** sum = 0;

**int** digits = *countDigits*(number);

**while** (number != 0) {

**int** digit = number % 10;

sum += *power*(digit, digits);

number /= 10;

}

**return** sum == originalNumber;

}

**public** **static** **int** countDigits(**int** number) {

**int** count = 0;

**while** (number != 0) {

count++;

number /= 10;

}

**return** count;

}

**public** **static** **int** power(**int** base, **int** exponent) {

**int** result = 1;

**for** (**int** i = 0; i < exponent; i++) {

result \*= base;

}

**return** result;

}

}

18. Write a program to check if a number is Neon Number or Not.

**package** org.ruchi;

**import** java.util.Scanner;

**public** **class** NeonNumber {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter a number: ");

**int** number = scanner.nextInt();

**int** square = number \* number;

**int** sumOfDigits = 0;

**while** (square != 0) {

sumOfDigits += square % 10;

square /= 10;

}

**if** (sumOfDigits == number) {

System.***out***.println(number + " is a Neon number.");

} **else** {

System.***out***.println(number + " is not a Neon number.");

}}}

19. Write a program to find the factorial of a given number.

**package** org.ruchi;

**import** java.util.Scanner;

**public** **class** Factorial {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

**int** fact = 1;

System.***out***.println("Enter a number: ");

**int** num = sc.nextInt();

**for**(**int** i = 1; i <= num; i++) {

fact \*= i;

}

System.***out***.println("The factorial of "+ num + " is " + fact);

}

}

20. Write a program to find the sum of Fibonacci Series numbers of first N even indexes.

**package** org.ruchi;

**import** java.util.Scanner;

**public** **class** FibonacciEvenSum {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter the value of N: ");

**int** N = sc.nextInt();

**int** a = 0, b = 1, sum = 0;

**int** count = 0;

**int** index = 0;

**while** (count < N) {

**if** (index % 2 == 0) {

sum += a;

count++;

}

**int** temp = a + b;

a = b;

b = temp;

index++;

}

System.***out***.println("Sum of first " + N + " Fibonacci numbers at even indexes: " + sum);

}}

21. Write a program to print right triangle star pattern.

**package** org.ruchi;

**import** java.util.Scanner;

**public** **class** RightTriangleStar {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter the number of rows: ");

**int** rows = sc.nextInt();

**for** (**int** i = 1; i <= rows; i++) {

**for** (**int** j = 1; j <= i; j++) {

System.***out***.print("\* ");

}

System.***out***.println();

}}}

22. Write a program to print reverse pyramid star pattern.

**package** org.ruchi;

**import** java.util.Scanner;

**public** **class** ReversePyramidStar {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the number of rows: ");

**int** rows = scanner.nextInt();

**for** (**int** i = rows; i >= 1; i--) {

**for** (**int** j = 1; j <= rows - i; j++) {

System.***out***.print(" ");

}

**for** (**int** k = 1; k <= 2 \* i - 1; k++) {

System.***out***.print("\*");

}

System.***out***.println();

}

}

}

23. Write a program to print upper star triangle pattern.

**package** org.ruchi;

**import** java.util.Scanner;

**public** **class** UpperTriangleStar {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the number of rows: ");

**int** rows = scanner.nextInt();

**for** (**int** i = 1; i <= rows; i++) {

**for** (**int** j = rows; j > i; j--) {

System.***out***.print(" ");

}

**for** (**int** k = 1; k <= i; k++) {

System.***out***.print("\* ");

}

System.***out***.println();

}}}

24. Write a program to print diamond shape star pattern.

**package** org.ruchi;

**import** java.util.Scanner;

**public** **class** DiamondStar{

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the number of rows: ");

**int** rows = scanner.nextInt();

**for** (**int** i = 1; i <= rows; i++) {

// Print leading spaces

**for** (**int** j = rows; j > i; j--) {

System.***out***.print(" "); }

**for** (**int** k = 1; k <= (2 \* i - 1); k++) {

System.***out***.print("\*");}

System.***out***.println();}

**for** (**int** i = rows - 1; i >= 1; i--) {

// Print leading spaces

**for** (**int** j = rows; j > i; j--) {

System.***out***.print(" ");

}

**for** (**int** k = 1; k <= (2 \* i - 1); k++) {

System.***out***.print("\*");

}

System.***out***.println();

}}}

25. Write a program to print square star pattern.

**package** org.ruchi;

**import** java.util.Scanner;

**public** **class** SquareStar {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the number of rows (and columns): ");

**int** rows = scanner.nextInt();

**for** (**int** i = 1; i <= rows; i++) {

**for** (**int** j = 1; j <= rows; j++) {

**if** (i == 1 || i == rows || j == 1 || j == rows) {

System.***out***.print("\* ");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}