CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

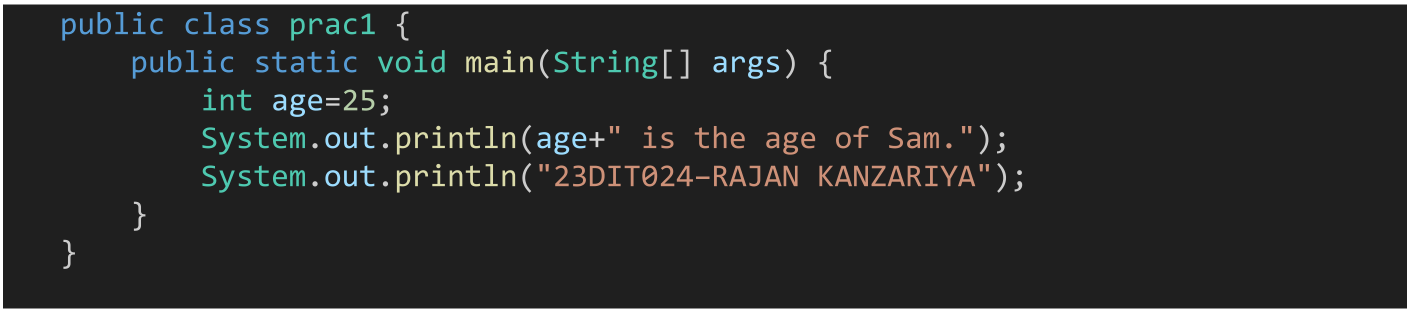
DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH

Department of Information Technology

PRACTICALS

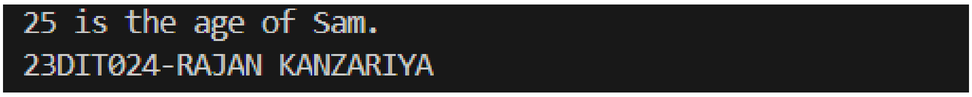
1) **AIM:** In a peaceful town, a budding programmer named Sam was tasked by her mentor, Ms. Java, to declare an integer variable named age, assign it the value of 25, and display it in a sentence. Sam quickly took to her computer and, with focus,wrote a program that would show "25 is the age of Sam." on the screen. Pleased with her work, Sam proudly presented her solution to Ms. Java, who commended her for her precise and clear coding skills.

CODE:



```java
public class prac1 {
    public static void main(String[] args) {
        int age=25;
        System.out.println(age+" is the age of Sam.");
        System.out.println("23DIT024-RAJAN KANZARIYA");
    }
}
```

OUTPUT:



```
25 is the age of Sam.
23DIT024-RAJAN KANZARIYA
```

CONCLUSION :

From this practical, I've learned how to declare an integer variable and assign a value to it. I practiced displaying the variable in a sentence using print statements. This helped me understand basic syntax and output functions in Java. It reinforced the importance of clear and precise coding.

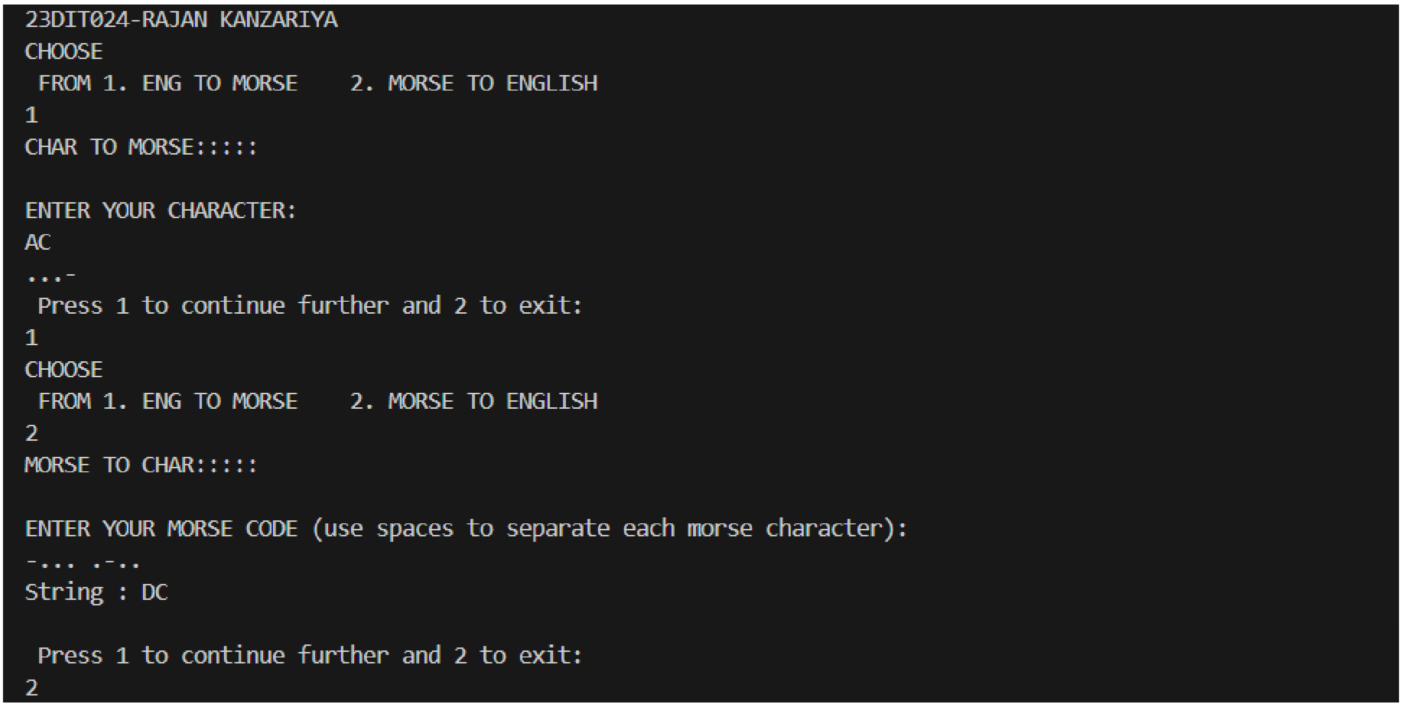
2) AIM : Java program that converts a string entered by the user to Morsecode or vice versa. It will require the implementation of data structures, including arrays, loops, and conditional statements.

- Create two arrays - one to contain the strings of letters to be converted and one to contain the Morse codes.
- In the program's main method, prompt the user for input to choose between the string or Morse.
- For Morse code conversion, read a string from the user; use conditional statements, looping, and array methods to convert the string to Morse-code.
- For string conversion, read in a Morse-coded string from the user; use arrays, conditional statements, and looping to convert
Morse code to a string

CODE:

```
     import java.util.*;
class prac2{
    public static void main(String[] args){
        char[] a={'A','B','C','D'};
        String[] b={"...-","..-.",".-..","-..."};
        int n;
        System.out.println("23DIT024-RAJAN KANZARIYA");

        do {
            System.out.println("CHOOSE \n FROM 1. ENG TO MORSE \t 2. MORSE TO
ENGLISH");
            Scanner obj = new Scanner(System.in);
            int x=obj.nextInt();

            if(x==1){
            System.out.println("CHAR TO MORSE::::: \n");
            System.out.println("ENTER YOUR CHARACTER: ");
            Scanner obj1 = new Scanner(System.in);
            String line=obj1.nextLine();

            for(int i=0;i<line.length();i++){
                char s = line.charAt(i);
                if(s==a[i]){
                    System.out.print(b[i]);
                }
             }
            }
            else {
            System.out.println("MORSE TO CHAR::::: \n");
            System.out.println("ENTER YOUR MORSE CODE (use spaces to separate
each morse character): ");
            Scanner obj2 = new Scanner(System.in);
            String line = obj2.nextLine();
            String[] s2 = line.split(" ");
            String cod = "";
            for (int i = 0; i < s2.length; i++) {
            int index = 0;
            for (int j = 0; j < b.length; j++) {
                if (b[j].equals(s2[i])) {
                    index = j;
                }
            }
            cod = cod + a[index];
            }
            System.out.println("String : " + cod);
            }
            System.out.println("\n Press 1 to continue further and 2 to exit:
");

            Scanner obj5= new Scanner(System.in);
            n=obj5.nextInt();
            } while (n!=2);

    }
}
```

OUTPUT :



```
23DIT024-RAJAN KANZARIYA
CHOOSE
 FROM 1. ENG TO MORSE    2. MORSE TO ENGLISH
1
CHAR TO MORSE:::::

ENTER YOUR CHARACTER:
AC
...-
 Press 1 to continue further and 2 to exit:
1
CHOOSE
 FROM 1. ENG TO MORSE    2. MORSE TO ENGLISH
2
MORSE TO CHAR:::::

ENTER YOUR MORSE CODE (use spaces to separate each morse character):
-... .-..
String : DC

 Press 1 to continue further and 2 to exit:
2
```

CONCLUSION :

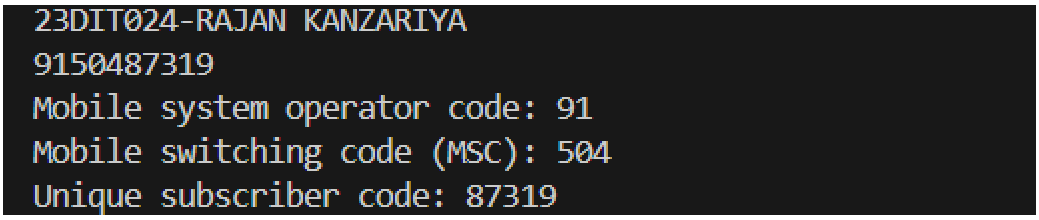
From this practical, I've learned how to implement data structures like arrays to store letter and Morse code mappings. I used loops and conditional statements for conversions, enhancing my skills in reading user input and conditional logic. This exercise demonstrated the complexity of string manipulation.

3) AIM : A typical mobile number in India is “+91-AA-BBB-CCCCC”. Where the first two digits (AA)indicate a mobile system operator, the next three (BBB) denote the mobile switching code(MSC) while the remaining five digits (CCCCC) are unique to the subscriber. Write an application that takes a mobile number as an input from a user in above mentioned format and display code for mobile system operator, mobile switching code and last 5 digits which are unique to subscriber. Ex. For an input +91-94-999-65789, output should be :Mobilesystem operator code is 94 MSC is 999 Unique code is 65789

CODE:

```
import java.util.Scanner;

class prac3 {
    public static void main(String[] args) {
        System.out.println("23DIT024-RAJAN KANZARIYA");

        Scanner obj = new Scanner(System.in);
        String mob = obj.nextLine();

        if (mob.length() == 10) {
            String mobileOperatorCode = mob.substring(0, 2);
            String mobileSwitchingCode = mob.substring(2, 5);
            String uniqueSubscriberCode = mob.substring(5);

            System.out.println("Mobile system operator code: " +
mobileOperatorCode);
            System.out.println("Mobile switching code (MSC): " +
mobileSwitchingCode);
            System.out.println("Unique subscriber code: " +
uniqueSubscriberCode);


        } else {
            System.out.println("Invalid input length. Please enter a 10-digit
mobile number.");
        }
    }
}
```

OUTPUT :



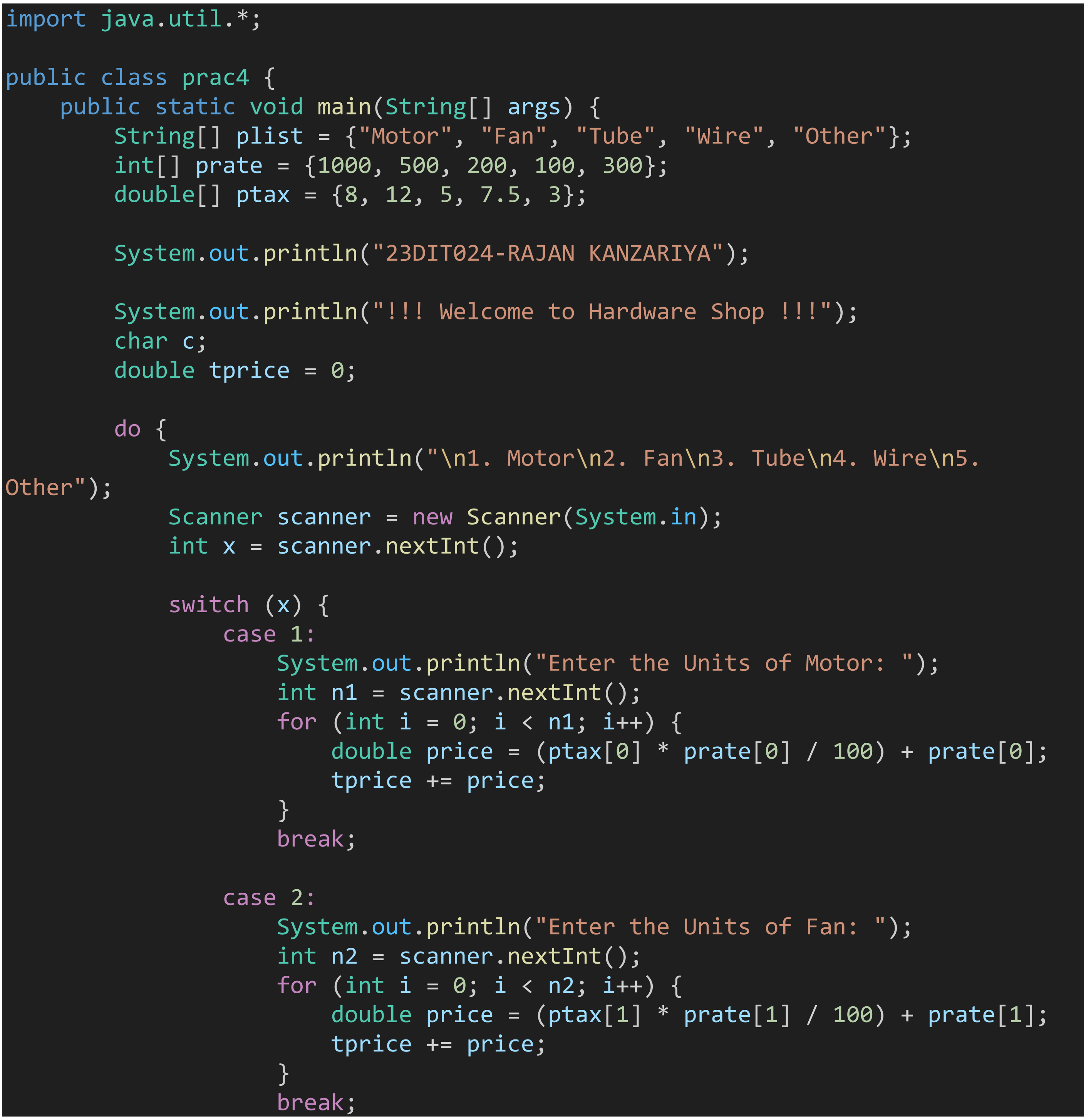
```
23DIT024-RAJAN KANZARIYA
9150487319
Mobile system operator code: 91
Mobile switching code (MSC): 504
Unique subscriber code: 87319
```

CONCLUSION:

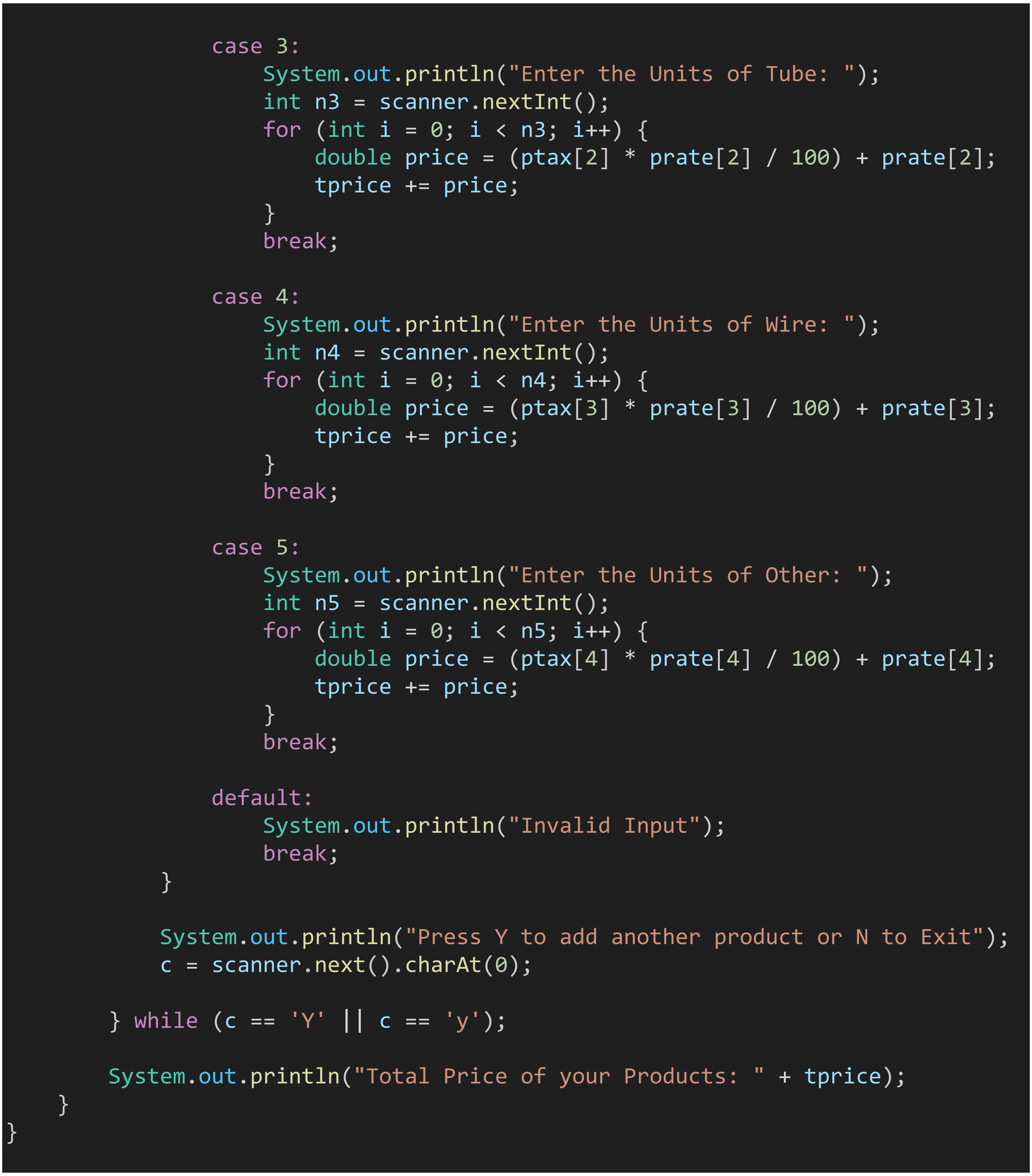
From this practical, I've learned how to write a program to parse a mobile number into its components. I practiced using string manipulation methods to extract and display specific parts, which helped me understand how to handle and validate formatted input strings. This exercise was essential for understanding data extraction from structured input.

4) AIM : An electric appliance shop assigns code 1 to motor,2 to fan,3 to tube and 4 for wires. All other items have code 5 or more. While selling the goods, a sales tax of 8% to motor,12% to fan,5% to tube light,7.5% to wires and 3% for all other items is charged. A list containing the product code and price in two different arrays. Write a java program using switch statement to prepare the bill.

CODE :

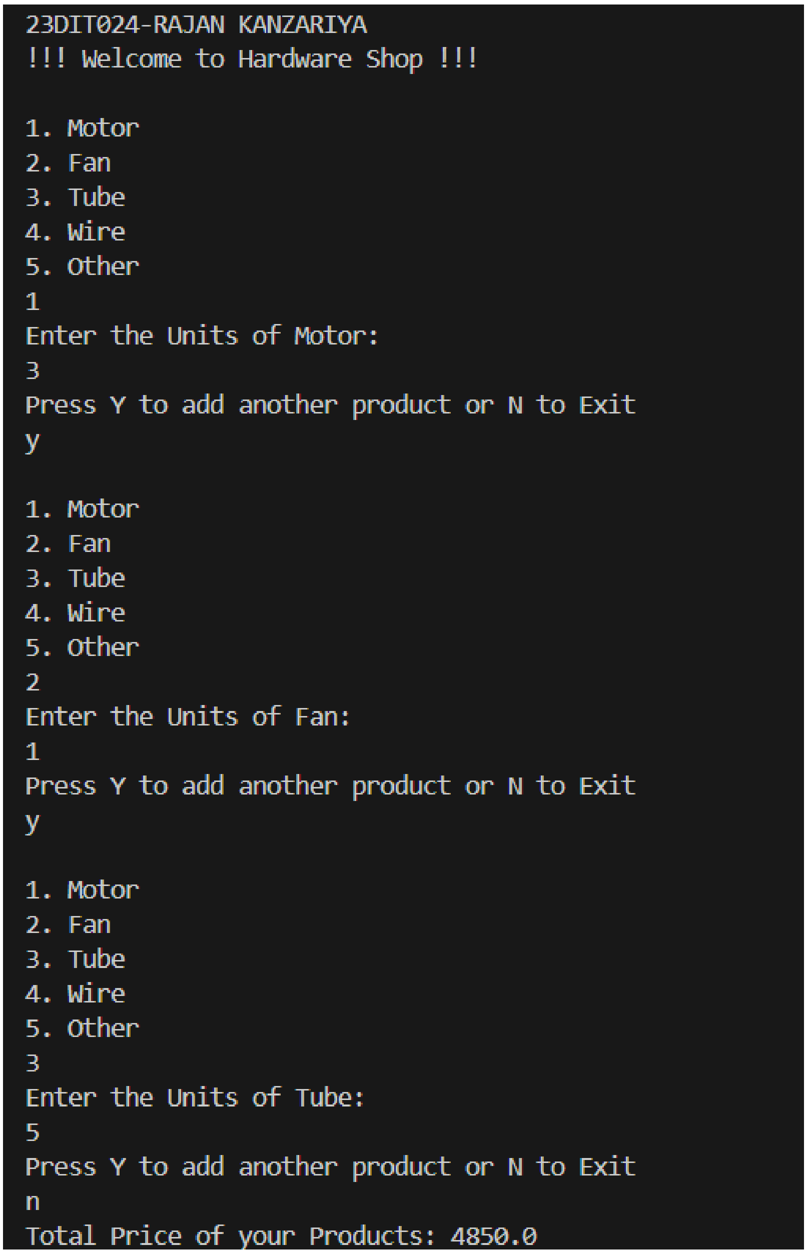


```
import java.util.*;

public class prac4 {
    public static void main(String[] args) {
        String[] plist = {"Motor", "Fan", "Tube", "Wire", "Other"};
        int[] prate = {1000, 500, 200, 100, 300};
        double[] ptax = {8, 12, 5, 7.5, 3};

        System.out.println("23DIT024-RAJAN KANZARIYA");

        System.out.println("!!! Welcome to Hardware Shop !!!");
        char c;
        double tprice = 0;

        do {
            System.out.println("\n1. Motor\n2. Fan\n3. Tube\n4. Wire\n5.
Other");
            Scanner scanner = new Scanner(System.in);
            int x = scanner.nextInt();

            switch (x) {
                case 1:
                    System.out.println("Enter the Units of Motor: ");
                    int n1 = scanner.nextInt();
                    for (int i = 0; i < n1; i++) {
                        double price = (ptax[0] * prate[0] / 100) + prate[0];
                        tprice += price;
                    }
                    break;

                case 2:
                    System.out.println("Enter the Units of Fan: ");
                    int n2 = scanner.nextInt();
                    for (int i = 0; i < n2; i++) {
                        double price = (ptax[1] * prate[1] / 100) + prate[1];
                        tprice += price;
                    }
                    break;
```



```
                case 3:
                    System.out.println("Enter the Units of Tube: ");
                    int n3 = scanner.nextInt();
                    for (int i = 0; i < n3; i++) {
                        double price = (ptax[2] * prate[2] / 100) + prate[2];
                        tprice += price;
                    }
                    break;

                case 4:
                    System.out.println("Enter the Units of Wire: ");
                    int n4 = scanner.nextInt();
                    for (int i = 0; i < n4; i++) {
                        double price = (ptax[3] * prate[3] / 100) + prate[3];
                        tprice += price;
                    }
                    break;

                case 5:
                    System.out.println("Enter the Units of Other: ");
                    int n5 = scanner.nextInt();
                    for (int i = 0; i < n5; i++) {
                        double price = (ptax[4] * prate[4] / 100) + prate[4];
                        tprice += price;
                    }
                    break;

                default:
                    System.out.println("Invalid Input");
                    break;
            }

            System.out.println("Press Y to add another product or N to Exit");
            c = scanner.next().charAt(0);

        } while (c == 'Y' || c == 'y');

        System.out.println("Total Price of your Products: " + tprice);
    }
}
```

OUTPUT :



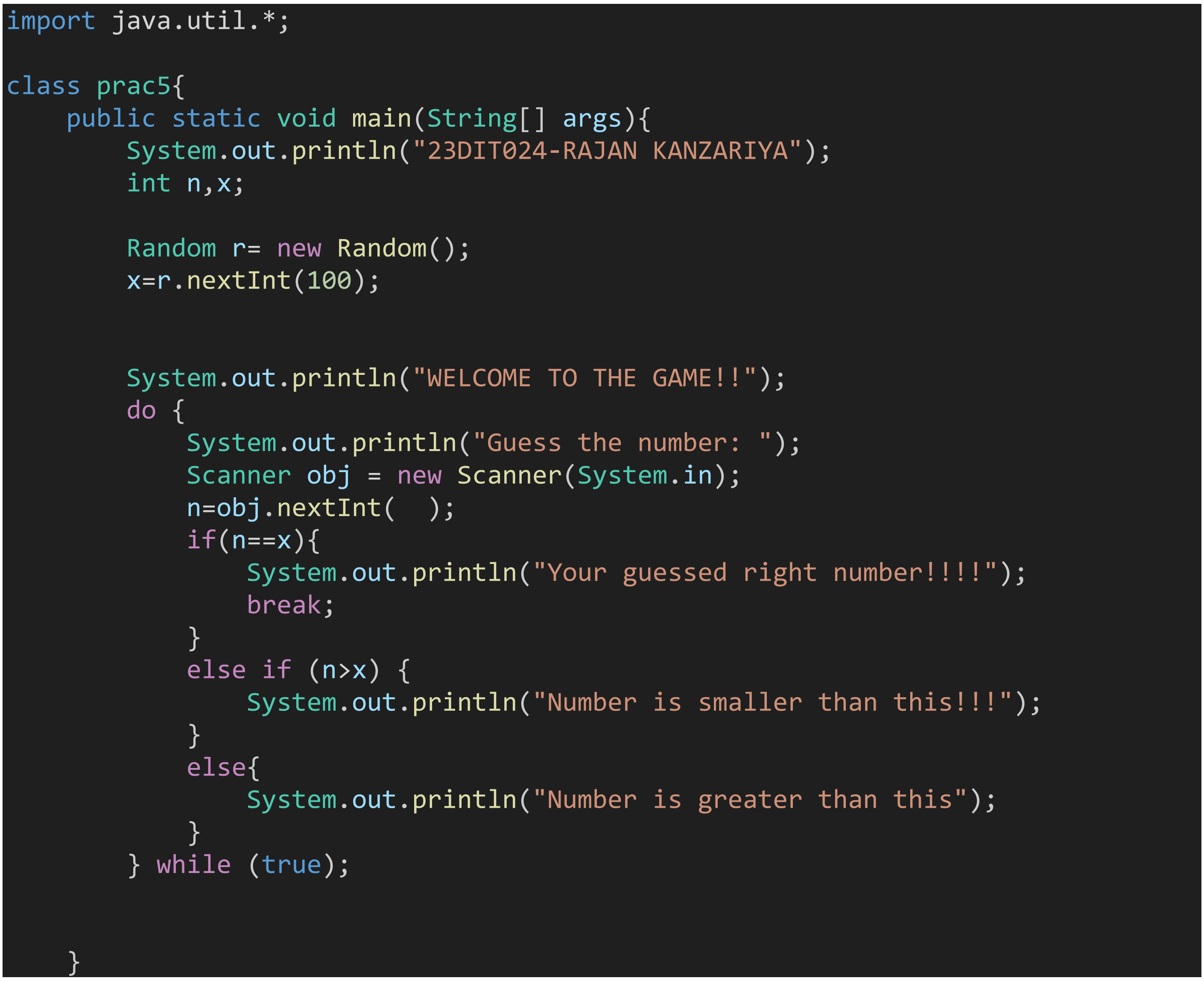
```
23DIT024-RAJAN KANZARIYA
!!! Welcome to Hardware Shop !!!

1. Motor
2. Fan
3. Tube
4. Wire
5. Other
1
Enter the Units of Motor:
3
Press Y to add another product or N to Exit
y

1. Motor
2. Fan
3. Tube
4. Wire
5. Other
2
Enter the Units of Fan:
1
Press Y to add another product or N to Exit
y

1. Motor
2. Fan
3. Tube
4. Wire
5. Other
3
Enter the Units of Tube:
5
Press Y to add another product or N to Exit
n
Total Price of your Products: 4850.0
```

CONCLUSION :

From this practical, I've learned to use switch statements to apply different tax rates based on product codes. I practiced creating arrays to store product codes and prices and calculating sales tax accordingly. This reinforced my understanding of control structures and array handling in Java.

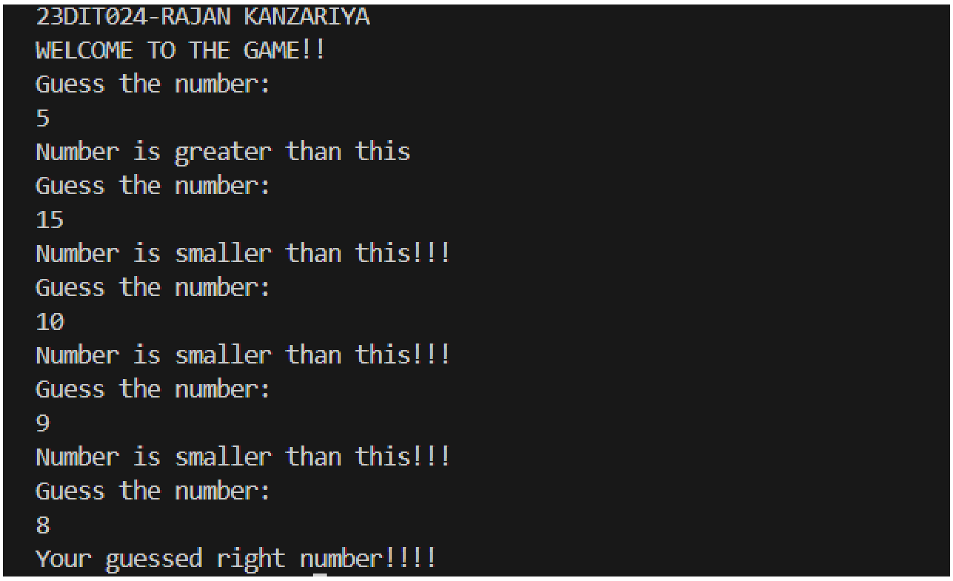
5) AIM : Create a Java program that simulates a guessing game,where the computer picks a random number between 1 and 100 and the user has to guess it. We can use the Scanner class to 1 get user input and a loop to allow multiple guesses.

CODE:



```
import java.util.*;

class prac5{
    public static void main(String[] args){
        System.out.println("23DIT024-RAJAN KANZARIYA");
        int n,x;

        Random r= new Random();
        x=r.nextInt(100);


        System.out.println("WELCOME TO THE GAME!!");
        do {
            System.out.println("Guess the number: ");
            Scanner obj = new Scanner(System.in);
            n=obj.nextInt(  );
            if(n==x){
                System.out.println("Your guessed right number!!!!");
                break;
            }
            else if (n>x) {
                System.out.println("Number is smaller than this!!!");
            }
            else{
                System.out.println("Number is greater than this");
            }
        } while (true);


    }
```

OUTPUT :



```
23DIT024-RAJAN KANZARIYA
WELCOME TO THE GAME!!
Guess the number:
5
Number is greater than this
Guess the number:
15
Number is smaller than this!!!
Guess the number:
10
Number is smaller than this!!!
Guess the number:
9
Number is smaller than this!!!
Guess the number:
8
Your guessed right number!!!!
```

CONCLUSION :

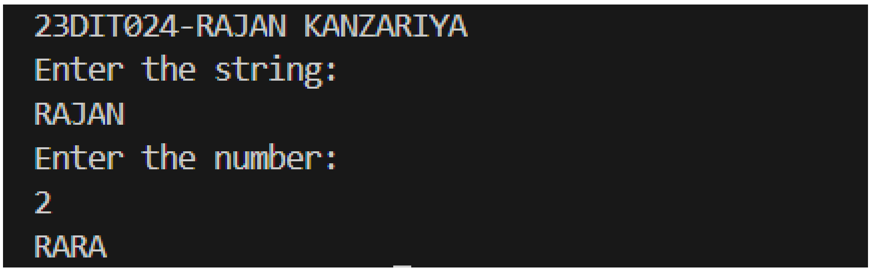
From this practical, I've learned how to develop a simple guessing game using loops and conditional statements. I practiced getting user input and providing feedback like "too low" or "too high." This exercise helped me understand the importance of loops and conditional logic in interactive programs.

6) AIM : Imagine you're tasked with creating a function that takes astring and a number. The goal is to repeat the first few characters of the string a specified number of times. If the string is shorter than the specified length, you should repeat whatever characters are available. How would you approach this problem?(function)

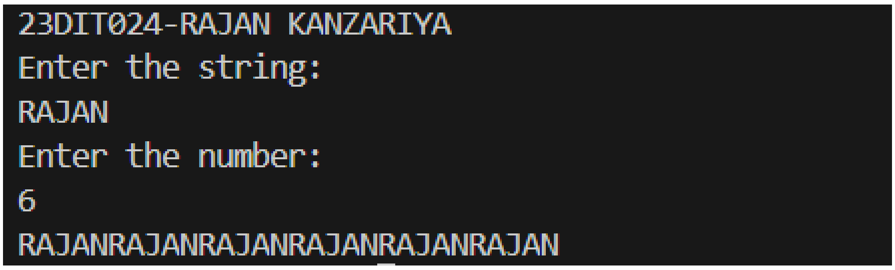
CODE:

```java
import java.util.Scanner;

public class prac6 {
    public static void main(String[] args){
        System.out.println("23DIT024-RAJAN KANZARIYA");
        System.out.println("Enter the string: ");
        Scanner obj = new Scanner(System.in);
        String n=obj.nextLine();

        System.out.println("Enter the number: ");
        Scanner obj2 = new Scanner(System.in);
        int x=obj2.nextInt(   );

        fun(x, n);
    }
    public static void fun(int x,String y){
        int a;
        String b;
        a=x;
        b=y;
        String c="",d="";
        if(y.length()<a){
            for(int i=0;i<a;i++){
                d=d+y;
            }
        }
        else{
        for(int i=0;i<a;i++){
            c=c+y.charAt(i);
        }
        for(int i=0;i<a;i++){
            d=d+c;
        }
    }
        System.out.println(d);
    }
}
```

OUTPUT :



```
23DIT024-RAJAN KANZARIYA
Enter the string:
RAJAN
Enter the number:
2
RARA
```



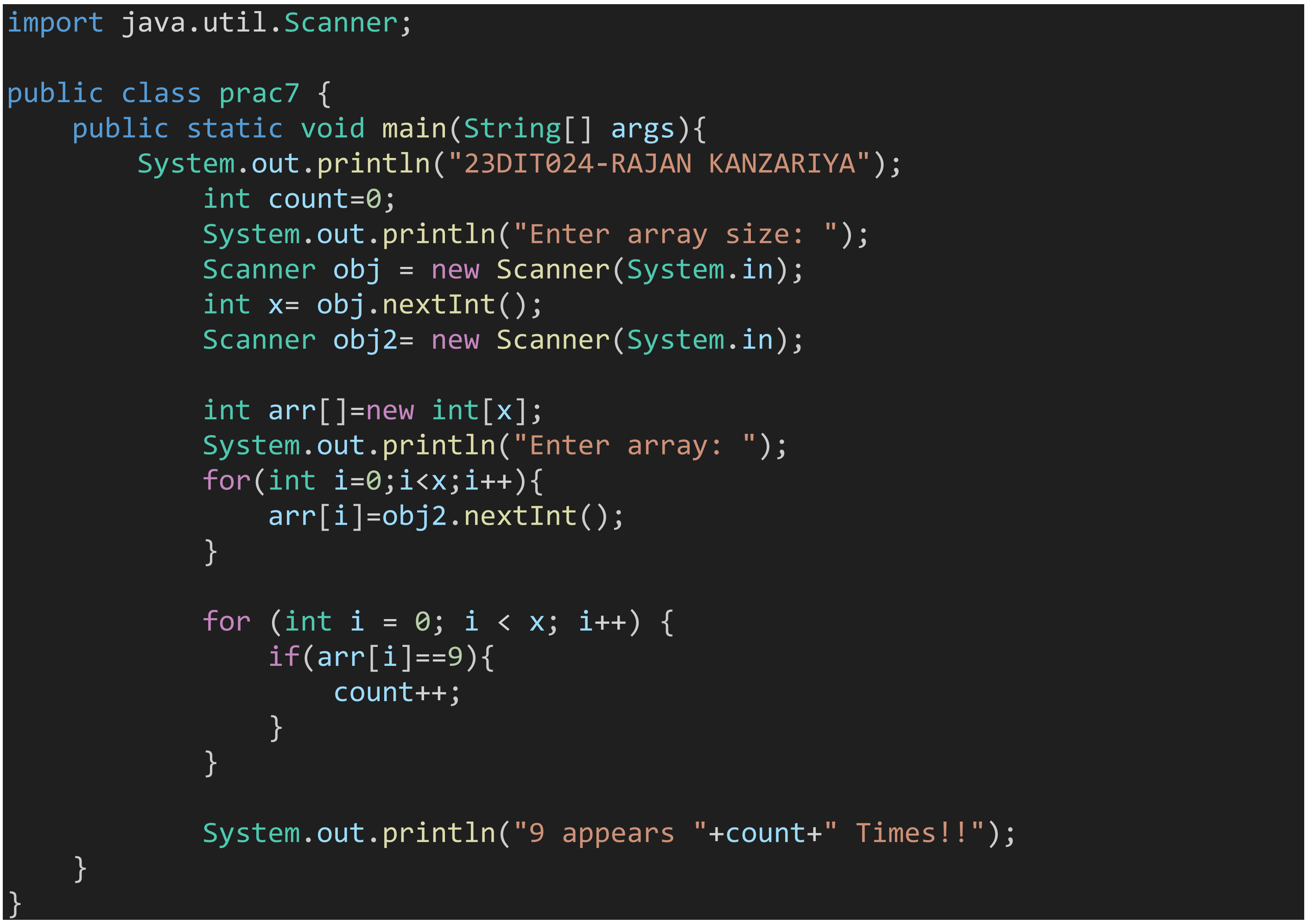
```
23DIT024-RAJAN KANZARIYA
Enter the string:
RAJAN
Enter the number:
6
RAJANRAJANRAJANRAJANRAJANRAJAN
```

CONCLUSION :

From this practical, I've learned how to write a function that repeats the first few characters of a string a specified number of times. I handled cases where the string is shorter than the specified length. This exercise enhanced my skills in string manipulation and function implementation.

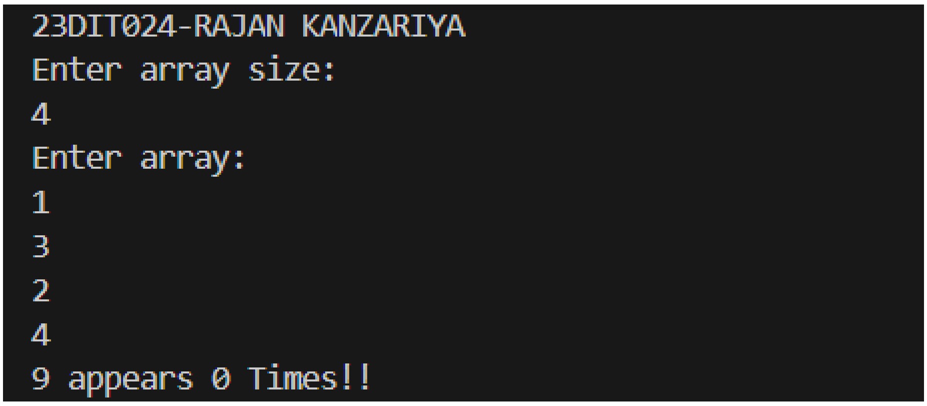
7) AIM : Imagine you're working with an array of integers, and your task is to count how many times the number 9 appears in the array. How would you write a Java program that efficiently determines this count, regardless of the array's size or the position of the numbers?

CODE:

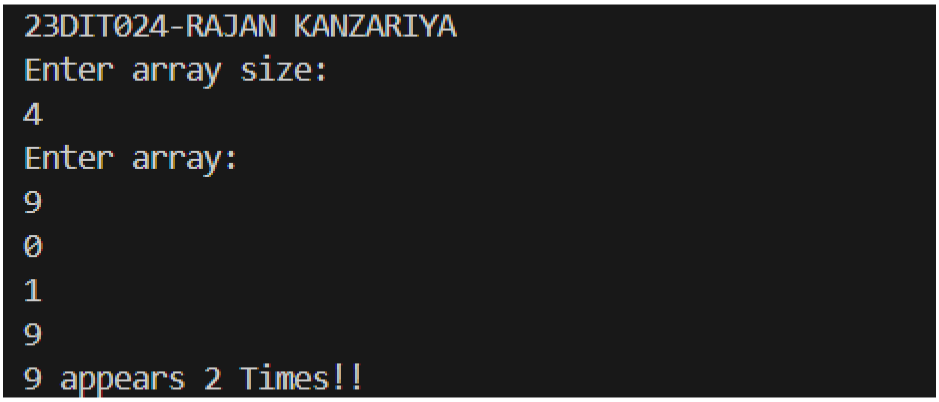


```java
import java.util.Scanner;

public class prac7 {
    public static void main(String[] args){
        System.out.println("23DIT024-RAJAN KANZARIYA");
            int count=0;
            System.out.println("Enter array size: ");
            Scanner obj = new Scanner(System.in);
            int x= obj.nextInt();
            Scanner obj2= new Scanner(System.in);

            int arr[]=new int[x];
            System.out.println("Enter array: ");
            for(int i=0;i<x;i++){
                arr[i]=obj2.nextInt();
            }

            for (int i = 0; i < x; i++) {
                if(arr[i]==9){
                    count++;
                }
            }

            System.out.println("9 appears "+count+" Times!!");
    }
}
```

OUTPUT :



```
23DIT024-RAJAN KANZARIYA
Enter array size:
4
Enter array:
1
3
2
4
9 appears 0 Times!!
```



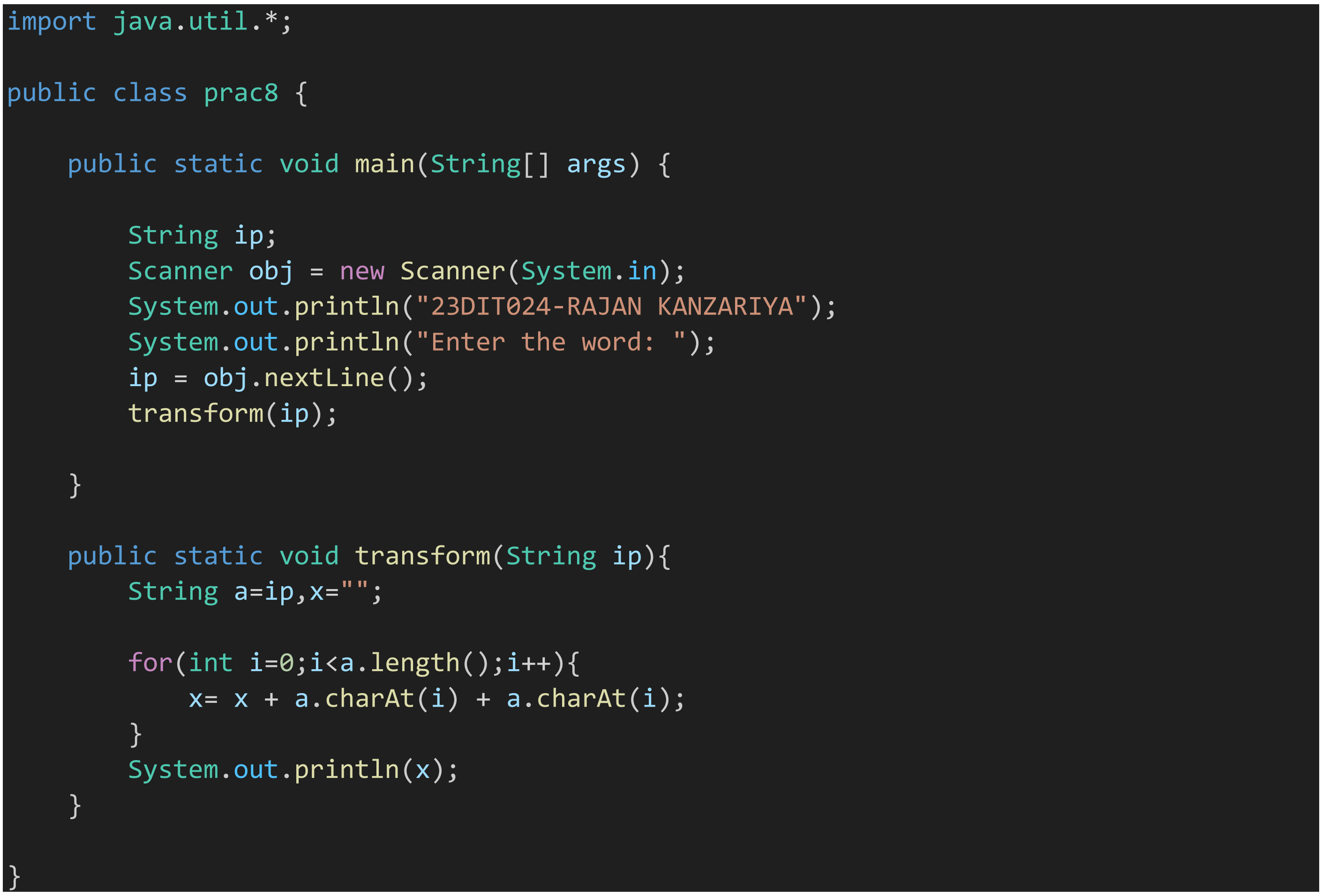
```
23DIT024-RAJAN KANZARIYA
Enter array size:
4
Enter array:
9
0
1
9
9 appears 2 Times!!
```

CONCLUSION:

From this practical, I've learned how to create a program that efficiently counts how many times a number appears in an array. I practiced using loops and conditional statements to traverse the array and count occurrences. This helped me understand efficient data processing techniques.

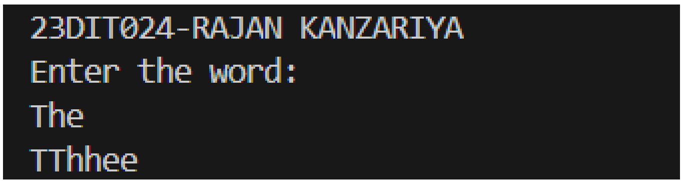
8) AIM : Suppose you are developing a text transformation tool. Your task is to create a function that takes a string and transforms it such that every character in the original string is doubled. For example, "The" becomes "TThhee". How would you design and implement this function in Java to handle any input string effectively?

CODE:



```java
import java.util.*;

public class prac8 {

    public static void main(String[] args) {

        String ip;
        Scanner obj = new Scanner(System.in);
        System.out.println("23DIT024-RAJAN KANZARIYA");
        System.out.println("Enter the word: ");
        ip = obj.nextLine();
        transform(ip);

    }

    public static void transform(String ip){
        String a=ip,x="";

        for(int i=0;i<a.length();i++){
            x= x + a.charAt(i) + a.charAt(i);
        }
        System.out.println(x);
    }

}
```

OUTPUT:



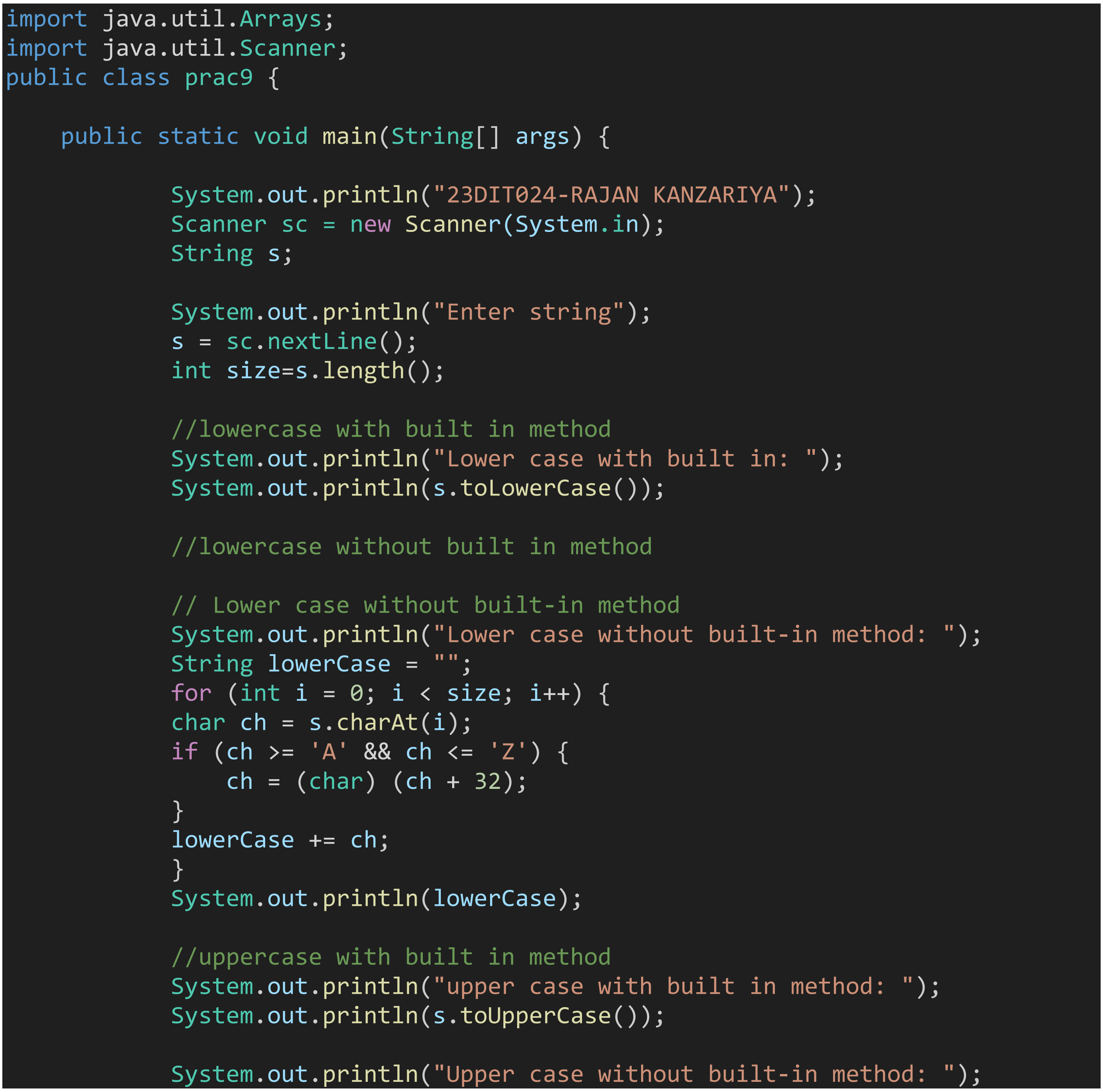
```
23DIT024-RAJAN KANZARIYA
Enter the word:
The
TThhee
```

CONCLUSION :

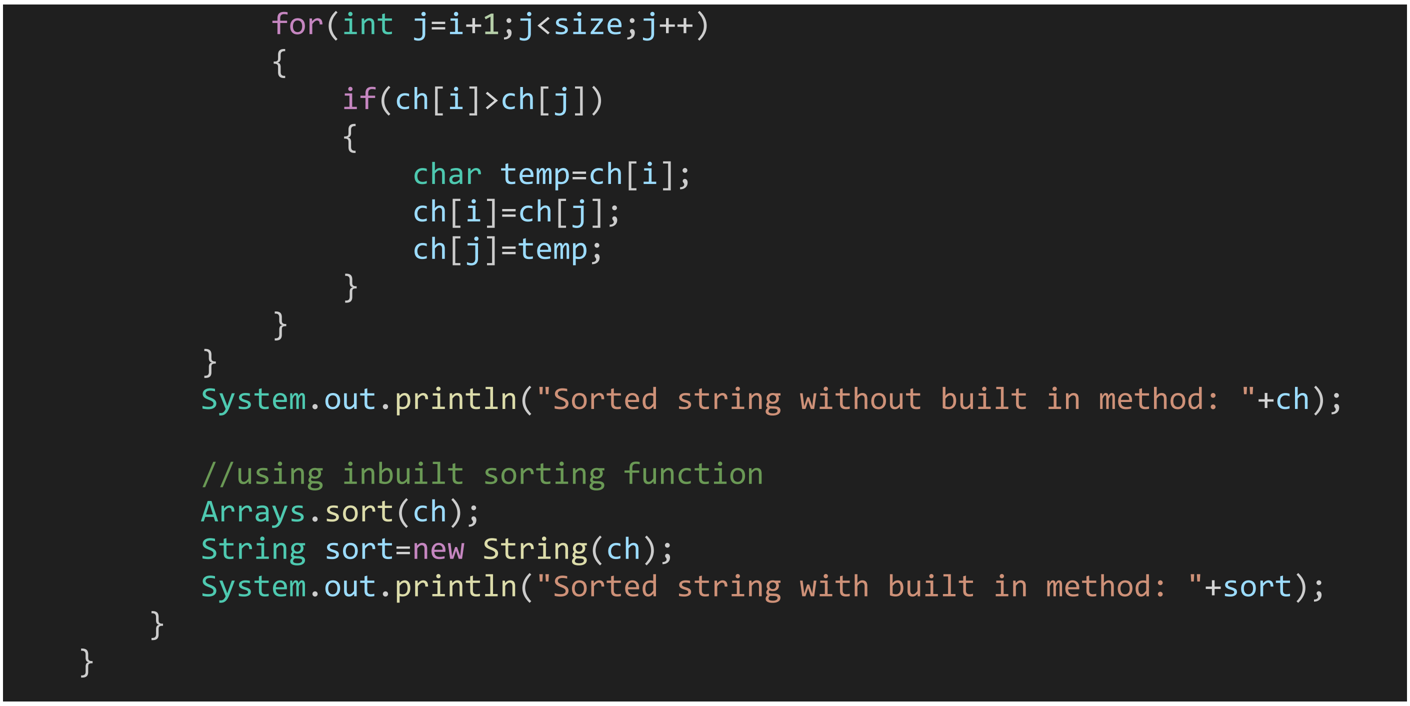
From this practical, I've learned how to implement a function to double each character in a string. I practiced using loops and string manipulation methods. This exercise improved my ability to work with strings and understand the importance of character-level operations.

9) AIM : you're a cybersecurity analyst investigating a suspicious string of characters. You need to analyze it thoroughly to uncover any hidden patterns or anomalies. The number of characters in the string to understand its size, Standardize the string for case-insensitive comparisons, Highlight potential keywords or acronyms, and Identify palindromes or potential encryption methods. Sort the string: Analyze character distribution and frequency.

CODE:

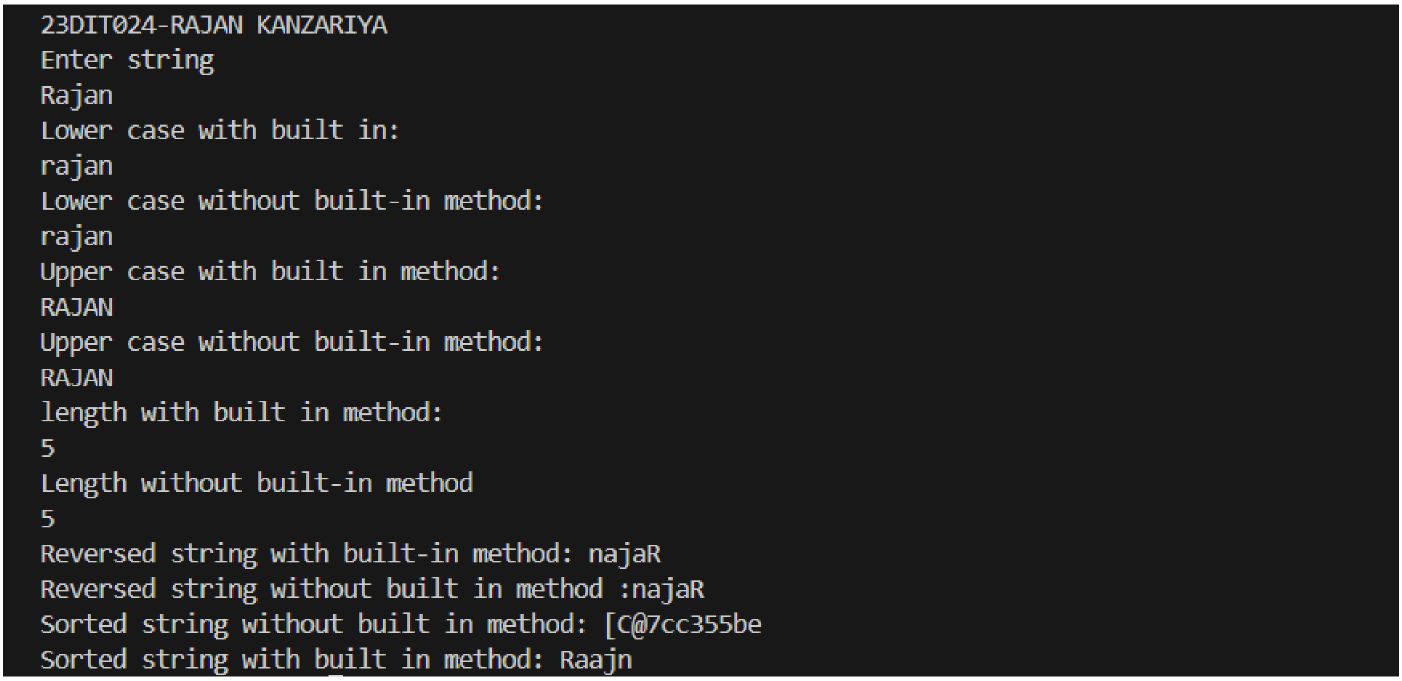


```java
import java.util.Arrays;
import java.util.Scanner;
public class prac9 {

    public static void main(String[] args) {

            System.out.println("23DIT024-RAJAN KANZARIYA");
            Scanner sc = new Scanner(System.in);
            String s;

            System.out.println("Enter string");
            s = sc.nextLine();
            int size=s.length();

            //lowercase with built in method
            System.out.println("Lower case with built in: ");
            System.out.println(s.toLowerCase());

            //lowercase without built in method

            // Lower case without built-in method
            System.out.println("Lower case without built-in method: ");
            String lowerCase = "";
            for (int i = 0; i < size; i++) {
            char ch = s.charAt(i);
            if (ch >= 'A' && ch <= 'Z') {
                ch = (char) (ch + 32);
            }
            lowerCase += ch;
            }
            System.out.println(lowerCase);

            //uppercase with built in method
            System.out.println("upper case with built in method: ");
            System.out.println(s.toUpperCase());

            System.out.println("Upper case without built-in method: ");
            String upperCase = "";
            for (int i = 0; i < size; i++) {
            char ch = s.charAt(i);
            if (ch >= 'a' && ch <= 'z') {
                ch = (char) (ch - 32);
            }
            upperCase += ch;
            }
            System.out.println(upperCase);

            //length with built in method
            System.out.println("length with built in method: ");
            System.out.println(s.length());

            //length without built in method
            System.out.println("Length without built-in method");
            int a = 0;
            try {
            while (s.charAt(a) != '\0') {
                a++;
            }
            } catch (StringIndexOutOfBoundsException e) {
            }
            System.out.println(a);

            //reverse with built in method
            String reversedStr = new StringBuilder(s).reverse().toString();
            System.out.println("Reversed string with built-in method: " +
reversedStr);

            //reverse without built in method
            String str="";
            for(int i=size-1;i>=0;i--)
            {

            str=str+s.charAt(i);

            }
            System.out.println("Reversed string without built in method
:"+str);

          //converting to character array
           char ch[]=new char[size];
           for(int i=0;i<size;i++)
           {
            ch[i]=s.charAt(i);
           }

          // with out using inbuilt sorting function.....
           for(int i=0;i<size;i++)
           {
```



```
                for(int j=i+1;j<size;j++)
                {
                    if(ch[i]>ch[j])
                    {
                        char temp=ch[i];
                        ch[i]=ch[j];
                        ch[j]=temp;
                    }
                }
            }
            System.out.println("Sorted string without built in method: "+ch);

            //using inbuilt sorting function
            Arrays.sort(ch);
            String sort=new String(ch);
            System.out.println("Sorted string with built in method: "+sort);
        }
    }
```

OUTPUT :



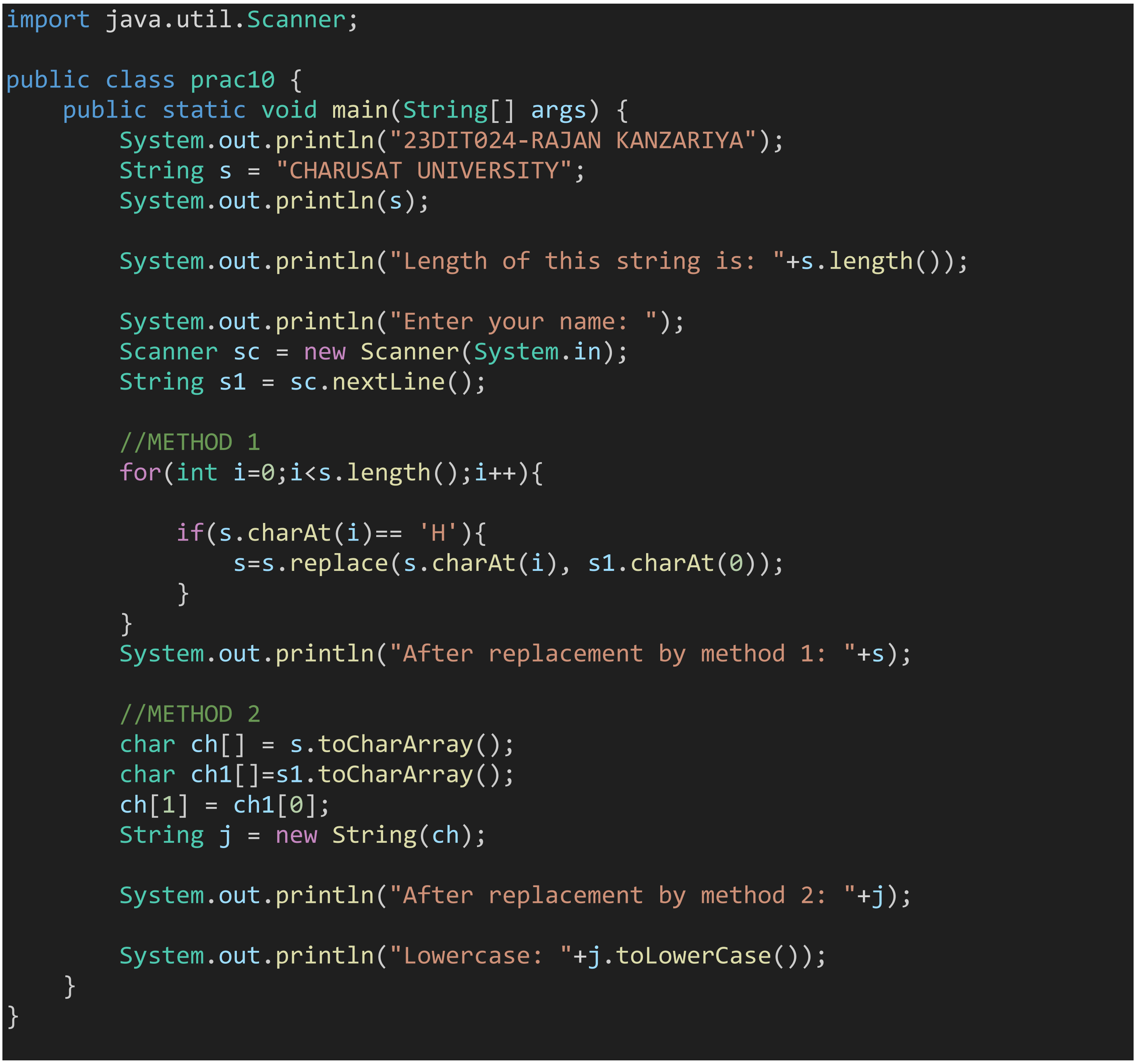
```
23DIT024-RAJAN KANZARIYA
Enter string
Rajan
Lower case with built in:
rajan
Lower case without built-in method:
rajan
Upper case with built in method:
RAJAN
Upper case without built-in method:
RAJAN
length with built in method:
5
Length without built-in method
5
Reversed string with built-in method: najaR
Reversed string without built in method :najaR
Sorted string without built in method: [C@7cc355be
Sorted string with built in method: Raajn
```

CONCLUSION :

From this practical, I've learned how to analyze a string for patterns, character frequency, and potential encryption methods. I practiced standardizing strings for case-insensitive comparisons and identifying palindromes. This exercise was crucial for understanding advanced string processing and analysis.

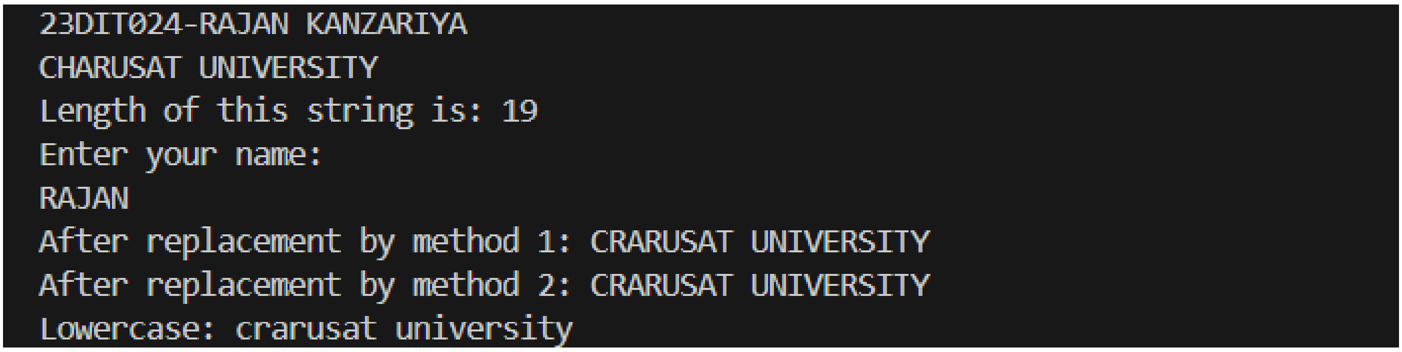
10) AIM : You're tasked with creating a basic encryption algorithm for your college project. The first step involves manipulating a given string, "CHARUSAT UNIVERSITY". Calculate the number of characters in the string to understand its structure, Identify the target character: The character to be replaced is 'H'. Replace the target character: Substitute 'H' with the first letter of your name. For instance, if your name starts with 'A', replace 'H' with 'A'. and Transform all characters to lowercase for consistency, and display the modified string.

CODE:



```java
import java.util.Scanner;

public class prac10 {
    public static void main(String[] args) {
        System.out.println("23DIT024-RAJAN KANZARIYA");
        String s = "CHARUSAT UNIVERSITY";
        System.out.println(s);

        System.out.println("Length of this string is: "+s.length());

        System.out.println("Enter your name: ");
        Scanner sc = new Scanner(System.in);
        String s1 = sc.nextLine();

        //METHOD 1
        for(int i=0;i<s.length();i++){

            if(s.charAt(i)== 'H'){
                s=s.replace(s.charAt(i), s1.charAt(0));
            }
        }
        System.out.println("After replacement by method 1: "+s);

        //METHOD 2
        char ch[] = s.toCharArray();
        char ch1[]=s1.toCharArray();
        ch[1] = ch1[0];
        String j = new String(ch);

        System.out.println("After replacement by method 2: "+j);

        System.out.println("Lowercase: "+j.toLowerCase());
    }
}
```

OUTPUT :



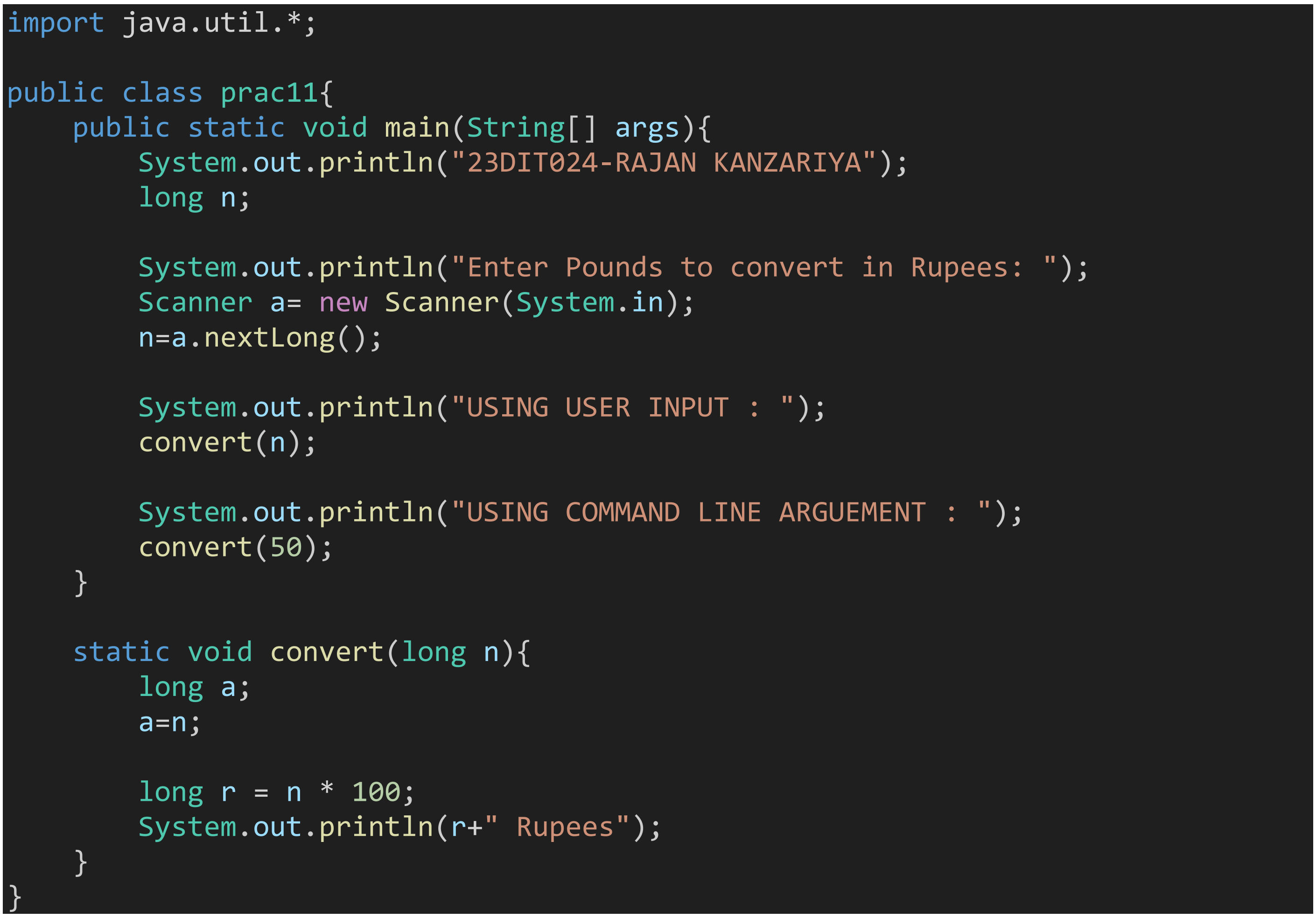
```
23DIT024-RAJAN KANZARIYA
CHARUSAT UNIVERSITY
Length of this string is: 19
Enter your name:
RAJAN
After replacement by method 1: CRARUSAT UNIVERSITY
After replacement by method 2: CRARUSAT UNIVERSITY
Lowercase: crarusat university
```

CONCLUSION :

From this practical, I've learned how to manipulate a string by replacing characters and transforming the case. I practiced calculating the number of characters in a string and making targeted substitutions. This exercise introduced me to basic encryption techniques and string manipulation.

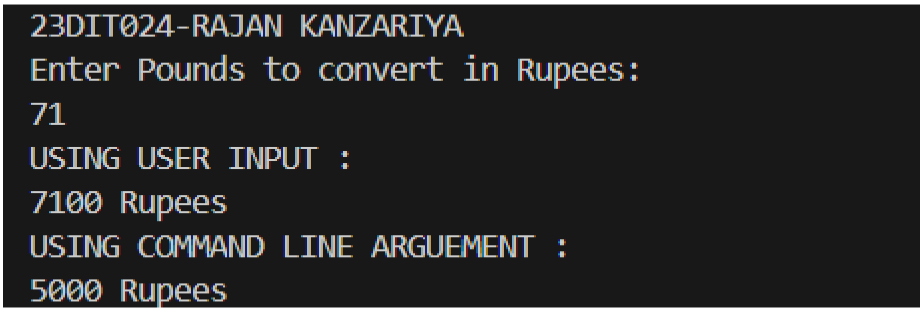
11) AIM : You're a budding Java programmer working on a currency conversion application. Your initial task is to convert Pounds to Rupees. To practice different input methods, you decide to implement two approaches: command-line arguments and user input using the Scanner class.

CODE:



```java
import java.util.*;

public class prac11{
    public static void main(String[] args){
        System.out.println("23DIT024-RAJAN KANZARIYA");
        long n;

        System.out.println("Enter Pounds to convert in Rupees: ");
        Scanner a= new Scanner(System.in);
        n=a.nextLong();

        System.out.println("USING USER INPUT : ");
        convert(n);

        System.out.println("USING COMMAND LINE ARGUEMENT : ");
        convert(50);
    }

    static void convert(long n){
        long a;
        a=n;

        long r = n * 100;
        System.out.println(r+" Rupees");
    }
}
```

OUTPUT :



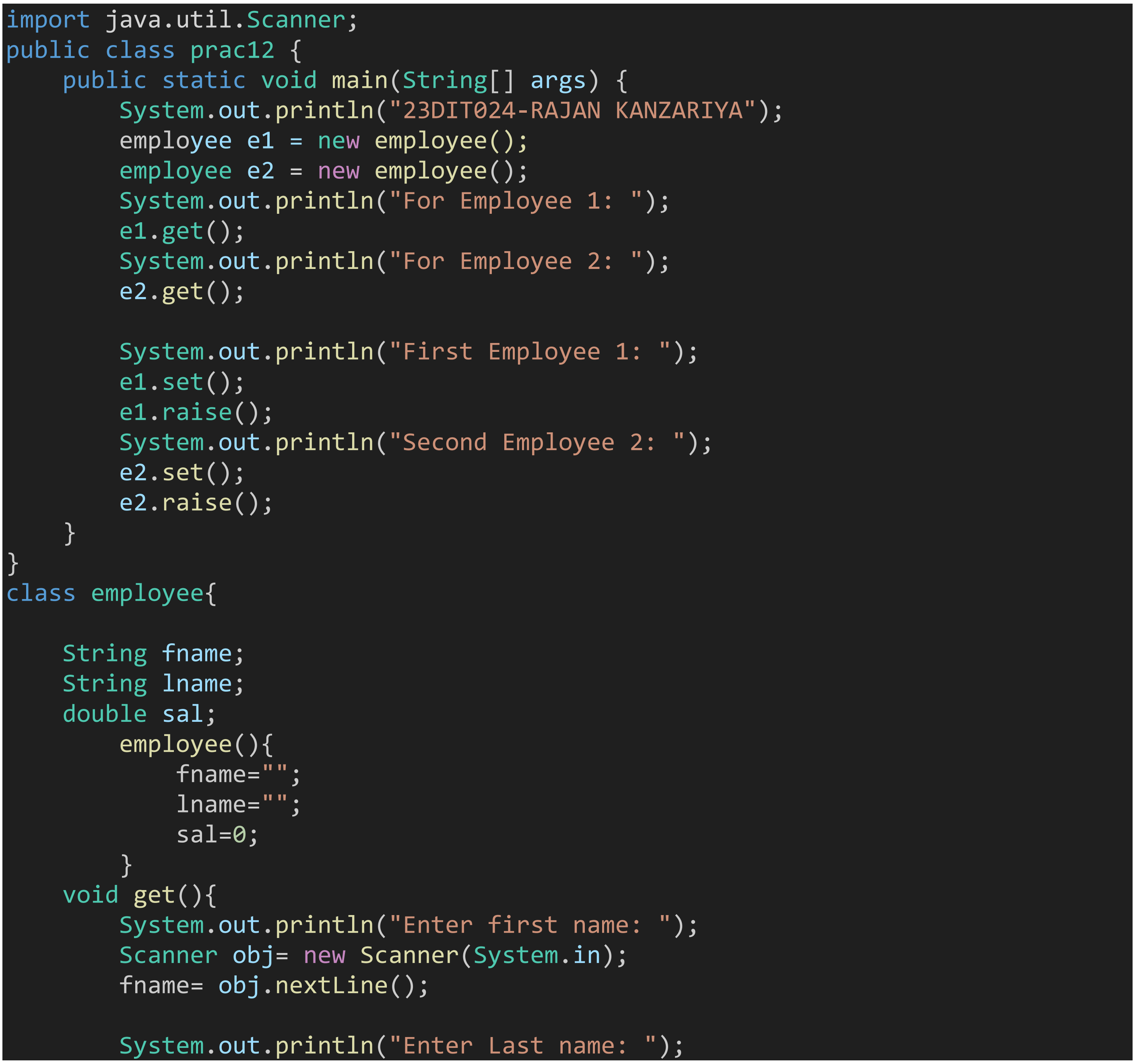
```
23DIT024-RAJAN KANZARIYA
Enter Pounds to convert in Rupees:
71
USING USER INPUT :
7100 Rupees
USING COMMAND LINE ARGUEMENT :
5000 Rupees
```

CONCLUSION :

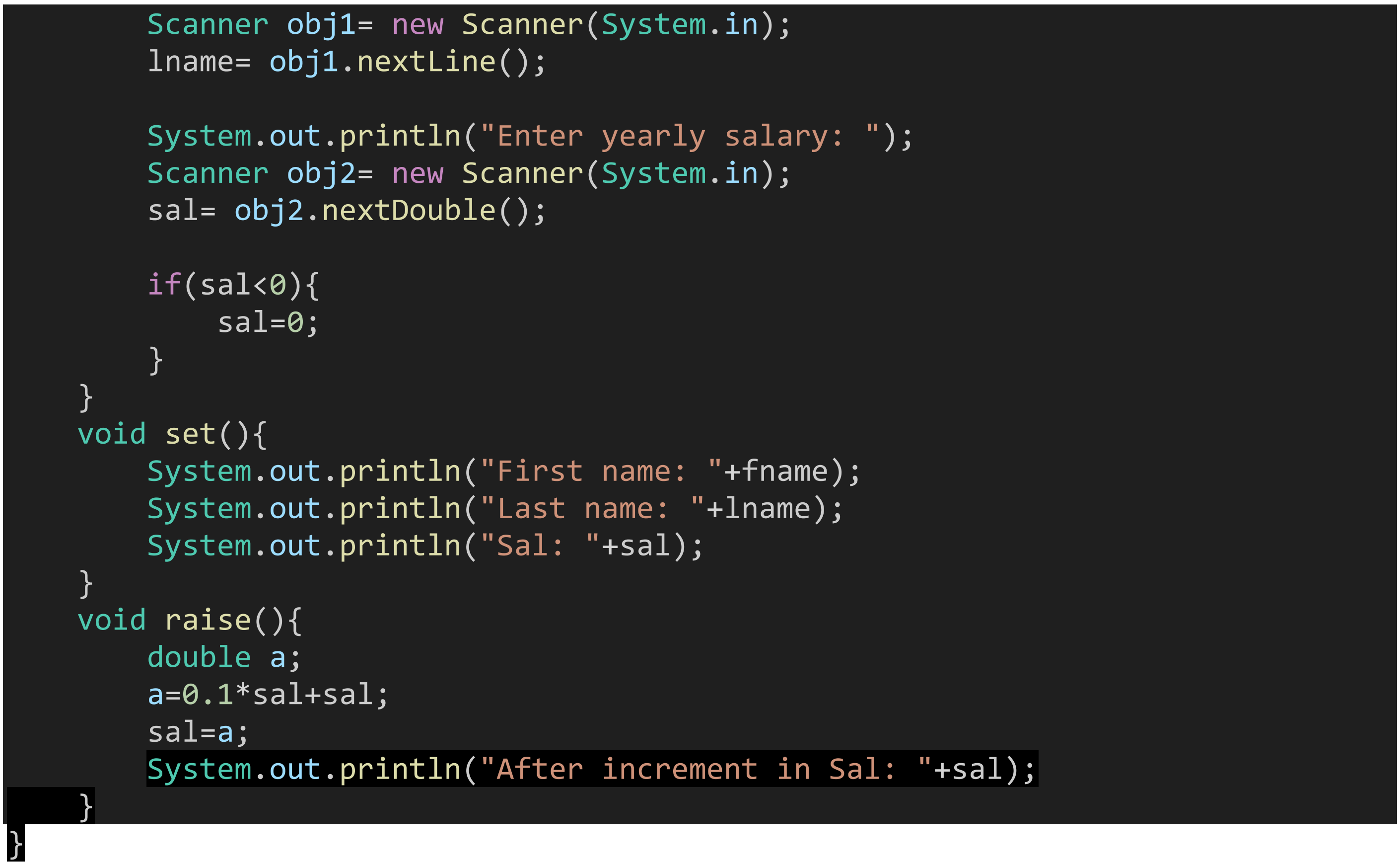
From this practical, I've learned how to implement currency conversion using command-line arguments and user input. I practiced using the Scanner class and handling different input methods. This helped me understand user input processing and basic arithmetic operations in Java.

12) AIM : Create a class called Employee that includes three pieces of information as instance variables—a first name (type String), a last name (type String), and a monthly salary (double). Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test application named EmployeeTest that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10% raise and display each Employee's yearly salary again.

CODE:

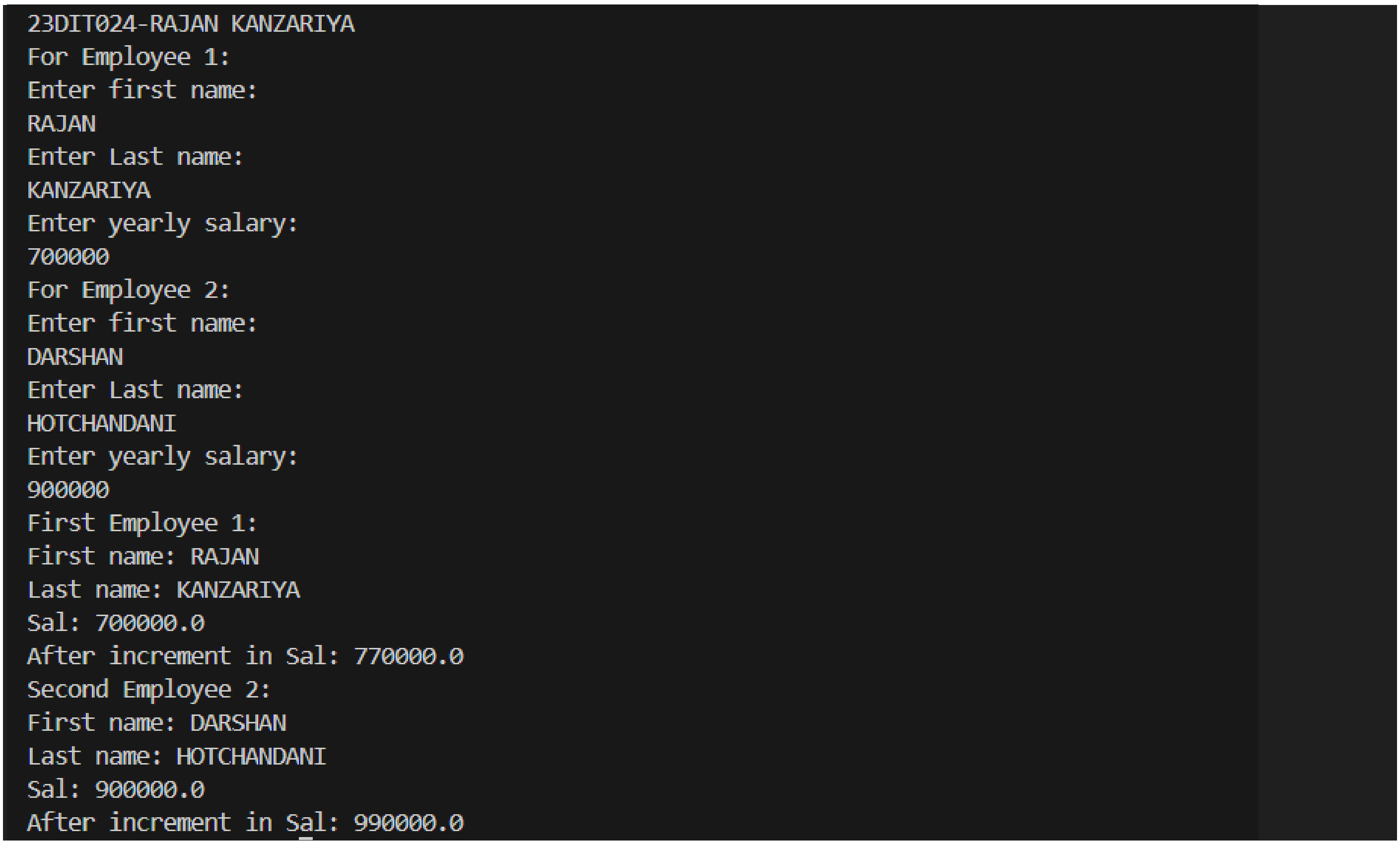


```
import java.util.Scanner;
public class prac12 {
    public static void main(String[] args) {
        System.out.println("23DIT024-RAJAN KANZARIYA");
        employee e1 = new employee();
        employee e2 = new employee();
        System.out.println("For Employee 1: ");
        e1.get();
        System.out.println("For Employee 2: ");
        e2.get();

        System.out.println("First Employee 1: ");
        e1.set();
        e1.raise();
        System.out.println("Second Employee 2: ");
        e2.set();
        e2.raise();
    }
}
class employee{

    String fname;
    String lname;
    double sal;
        employee(){
            fname="";
            lname="";
            sal=0;
        }
    void get(){
        System.out.println("Enter first name: ");
        Scanner obj= new Scanner(System.in);
        fname= obj.nextLine();

        System.out.println("Enter Last name: ");
```



```
        Scanner obj1= new Scanner(System.in);
        lname= obj1.nextLine();

        System.out.println("Enter yearly salary: ");
        Scanner obj2= new Scanner(System.in);
        sal= obj2.nextDouble();

        if(sal<0){
            sal=0;
        }
    }
    void set(){
        System.out.println("First name: "+fname);
        System.out.println("Last name: "+lname);
        System.out.println("Sal: "+sal);
    }
    void raise(){
        double a;
        a=0.1*sal+sal;
        sal=a;
        System.out.println("After increment in Sal: "+sal);
    }
}
```

OUTPUT :



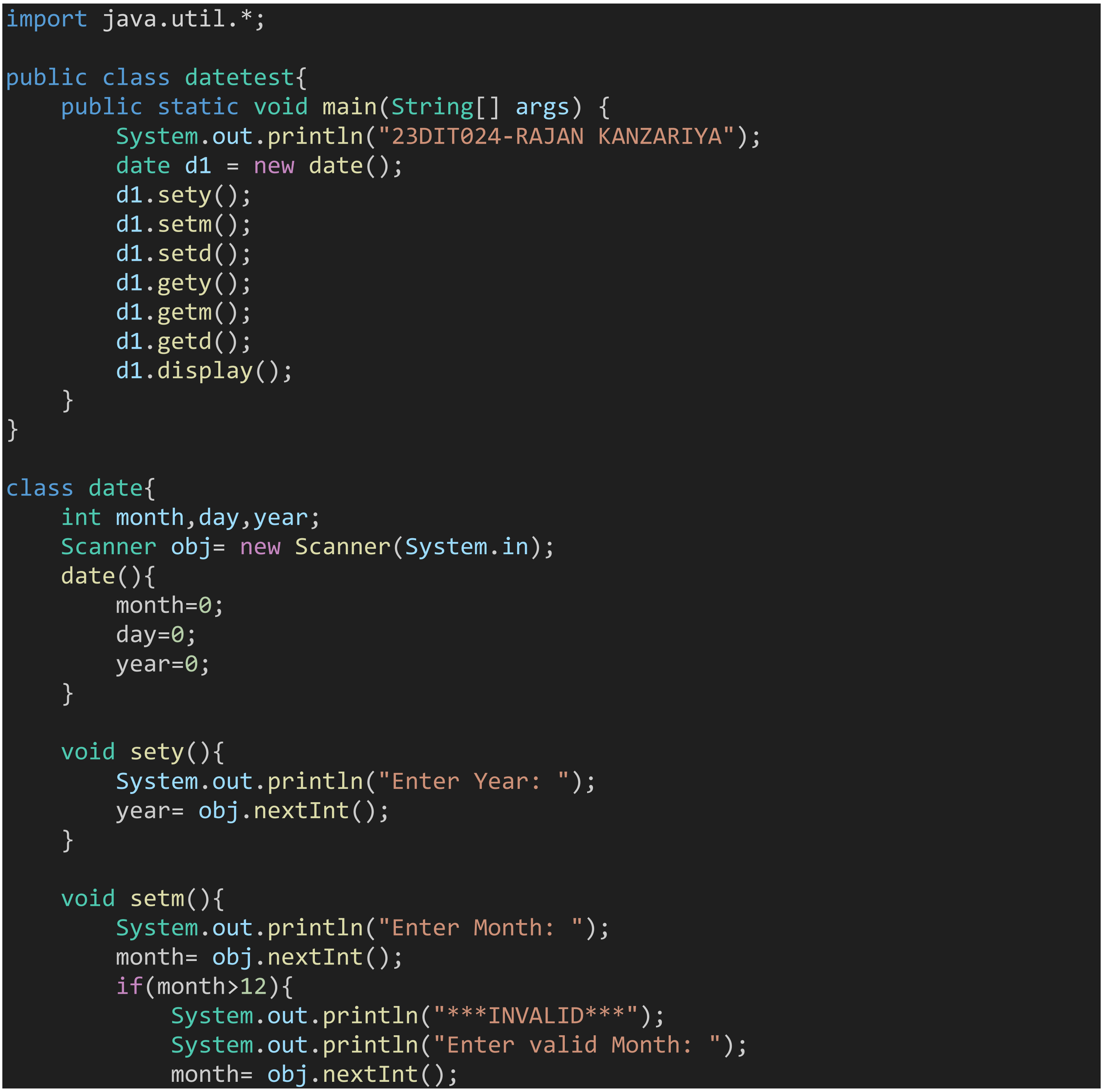
```
23DIT024-RAJAN KANZARIYA
For Employee 1:
Enter first name:
RAJAN
Enter Last name:
KANZARIYA
Enter yearly salary:
700000
For Employee 2:
Enter first name:
DARSHAN
Enter Last name:
HOTCHANDANI
Enter yearly salary:
900000
First Employee 1:
First name: RAJAN
Last name: KANZARIYA
Sal: 700000.0
After increment in Sal: 770000.0
Second Employee 2:
First name: DARSHAN
Last name: HOTCHANDANI
Sal: 900000.0
After increment in Sal: 990000.0
```

CONCLUSION :

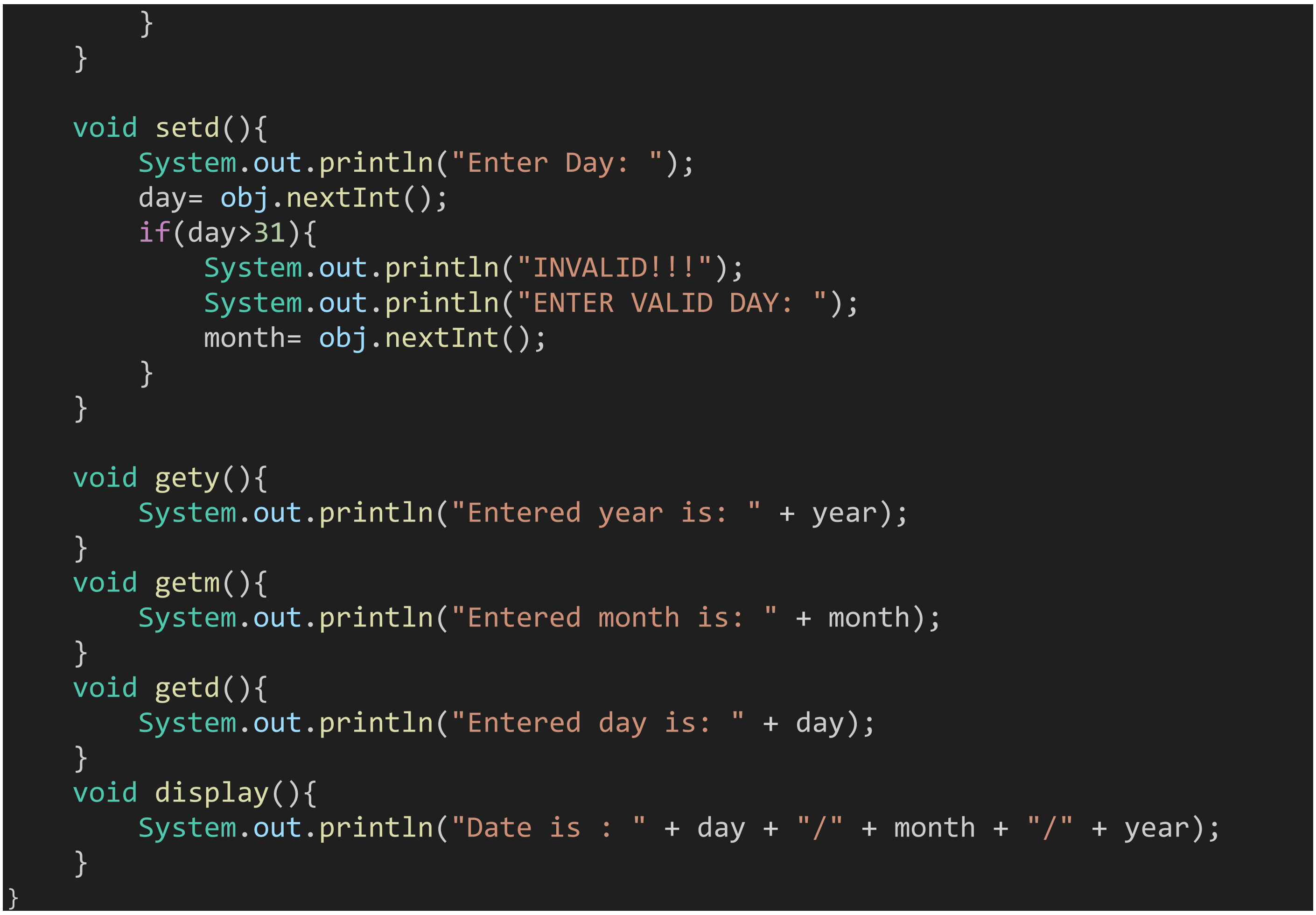
From this practical, I've learned how to create a class with instance variables, constructors, and methods for getting and setting values. I practiced ensuring data validation and manipulating object attributes. This exercise was essential for understanding object-oriented programming principles.

13) AIM : Create a class called Date that includes three pieces of information as instance variables—a month (type int), a day (type int) and a year (type int). Your class should have a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a method displayDate that displays the month, day and year separated by forward slashes (/). Write a test application named DateTest that demonstrates class Date's capabilities.

CODE:

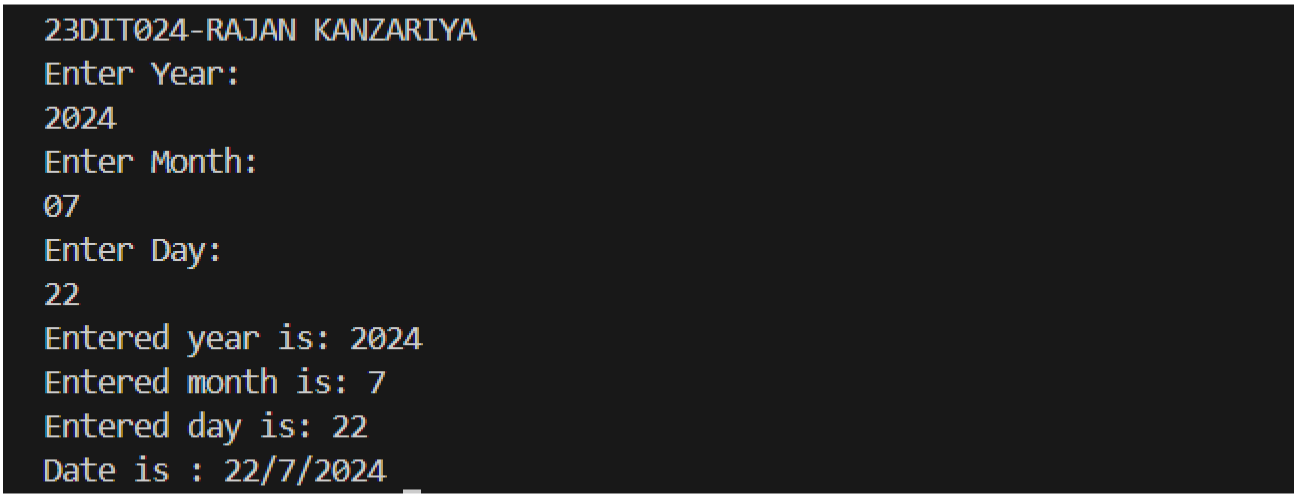


```
import java.util.*;

public class datetest{
    public static void main(String[] args) {
        System.out.println("23DIT024-RAJAN KANZARIYA");
        date d1 = new date();
        d1.sety();
        d1.setm();
        d1.setd();
        d1.gety();
        d1.getm();
        d1.getd();
        d1.display();
    }
}

class date{
    int month,day,year;
    Scanner obj= new Scanner(System.in);
    date(){
        month=0;
        day=0;
        year=0;
    }

    void sety(){
        System.out.println("Enter Year: ");
        year= obj.nextInt();
    }

    void setm(){
        System.out.println("Enter Month: ");
        month= obj.nextInt();
        if(month>12){
            System.out.println("***INVALID***");
            System.out.println("Enter valid Month: ");
            month= obj.nextInt();
```



```
        }
    }

    void setd(){
        System.out.println("Enter Day: ");
        day= obj.nextInt();
        if(day>31){
            System.out.println("INVALID!!!");
            System.out.println("ENTER VALID DAY: ");
            month= obj.nextInt();
        }
    }

    void gety(){
        System.out.println("Entered year is: " + year);
    }
    void getm(){
        System.out.println("Entered month is: " + month);
    }
    void getd(){
        System.out.println("Entered day is: " + day);
    }
    void display(){
        System.out.println("Date is : " + day + "/" + month + "/" + year);
    }
}
```

OUTPUT :



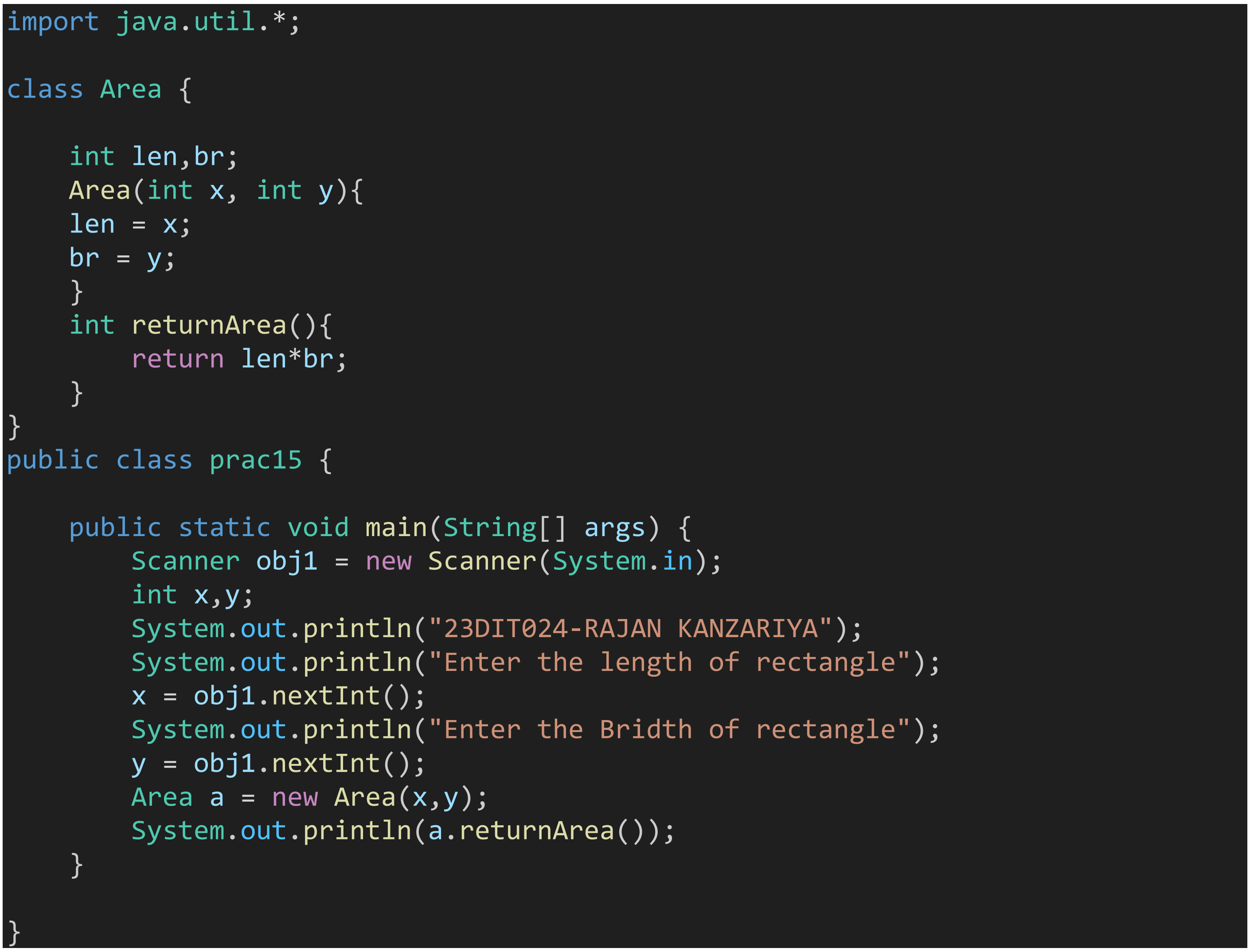
```
23DIT024-RAJAN KANZARIYA
Enter Year:
2024
Enter Month:
07
Enter Day:
22
Entered year is: 2024
Entered month is: 7
Entered day is: 22
Date is : 22/7/2024
```

CONCLUSION :

From this practical, I've learned how to create a class to handle dates and provide methods to display them. I practiced initializing instance variables through a constructor and ensuring data correctness. This reinforced my understanding of class design and data encapsulation.

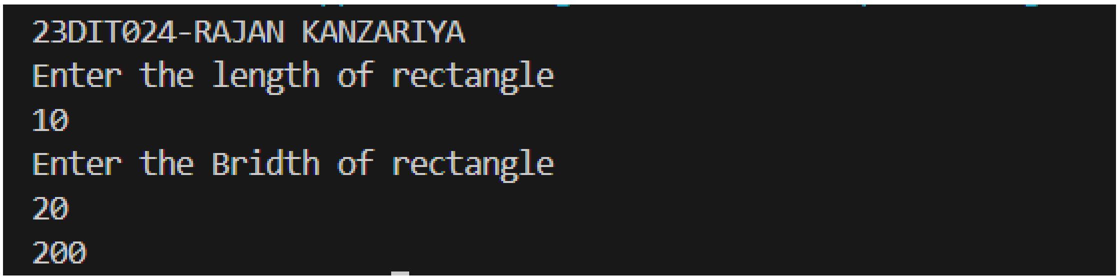
14) AIM : Write a program to print the area of a rectangle by creatin a class named 'Area' taking the values of its length and breadth as parameters of its constructor and having a method named 'returnArea' which returns the area of the rectangle. Length and breadth of rectangle are entered through keyboard.

CODE:



```java
import java.util.*;

class Area {

    int len,br;
    Area(int x, int y){
    len = x;
    br = y;
    }
    int returnArea(){
        return len*br;
    }
}
public class prac15 {

    public static void main(String[] args) {
        Scanner obj1 = new Scanner(System.in);
        int x,y;
        System.out.println("23DIT024-RAJAN KANZARIYA");
        System.out.println("Enter the length of rectangle");
        x = obj1.nextInt();
        System.out.println("Enter the Bridth of rectangle");
        y = obj1.nextInt();
        Area a = new Area(x,y);
        System.out.println(a.returnArea());
    }

}
```

OUTPUT :



```
23DIT024-RAJAN KANZARIYA
Enter the length of rectangle
10
Enter the Bridth of rectangle
20
200
```

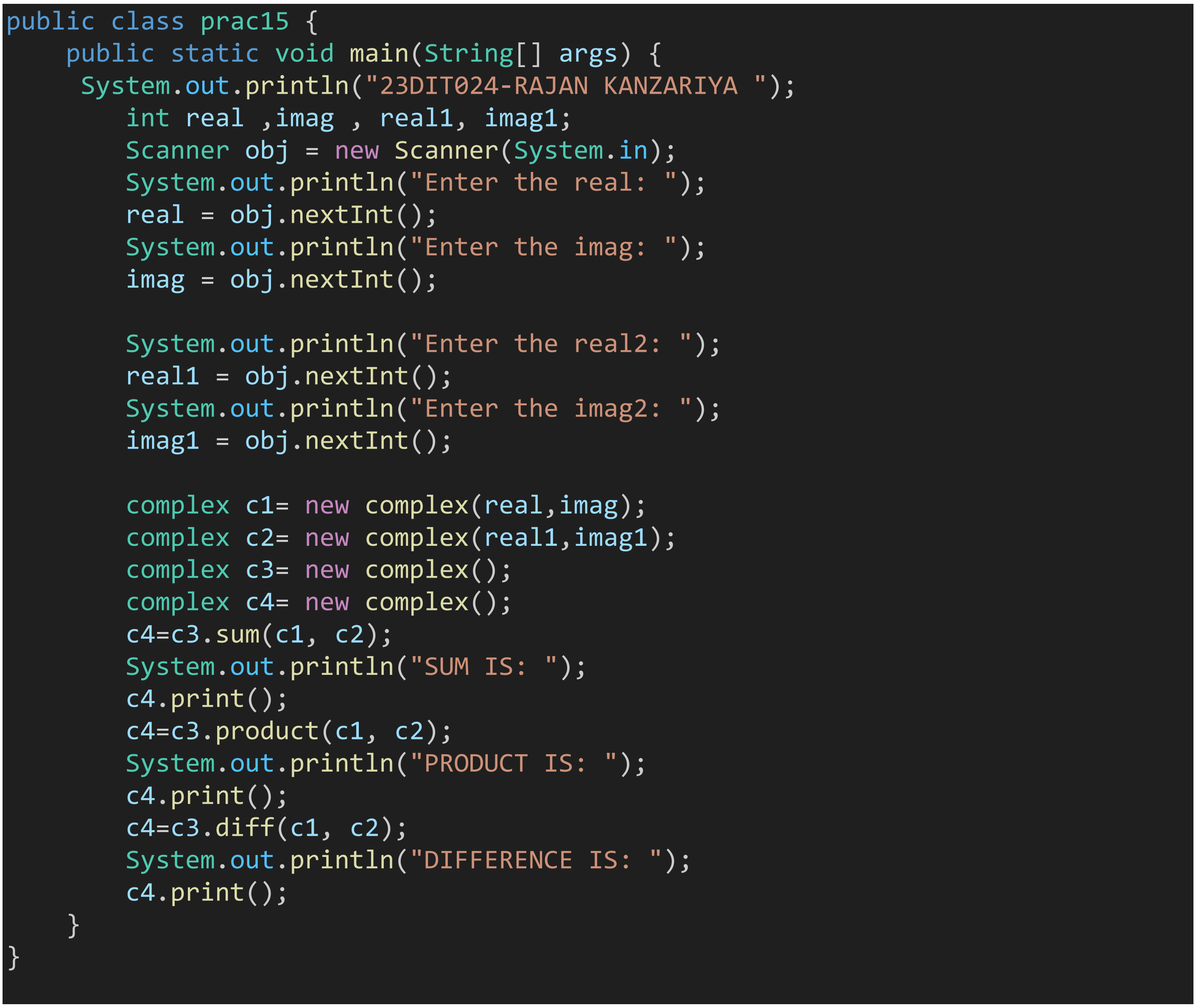
CONCLSUION :

From this practical, I've learned how to write a class to calculate and return the area of a rectangle using input values. I practiced creating constructors and methods to handle object attributes. This exercise was crucial for understanding object-oriented design and method implementation.

15) AIM : Imagine you're building a scientific calculator application. One crucial feature is handling complex numbers. You decide to create a Complex class to represent complex numbers and perform operations on them.(sum, difference and product)

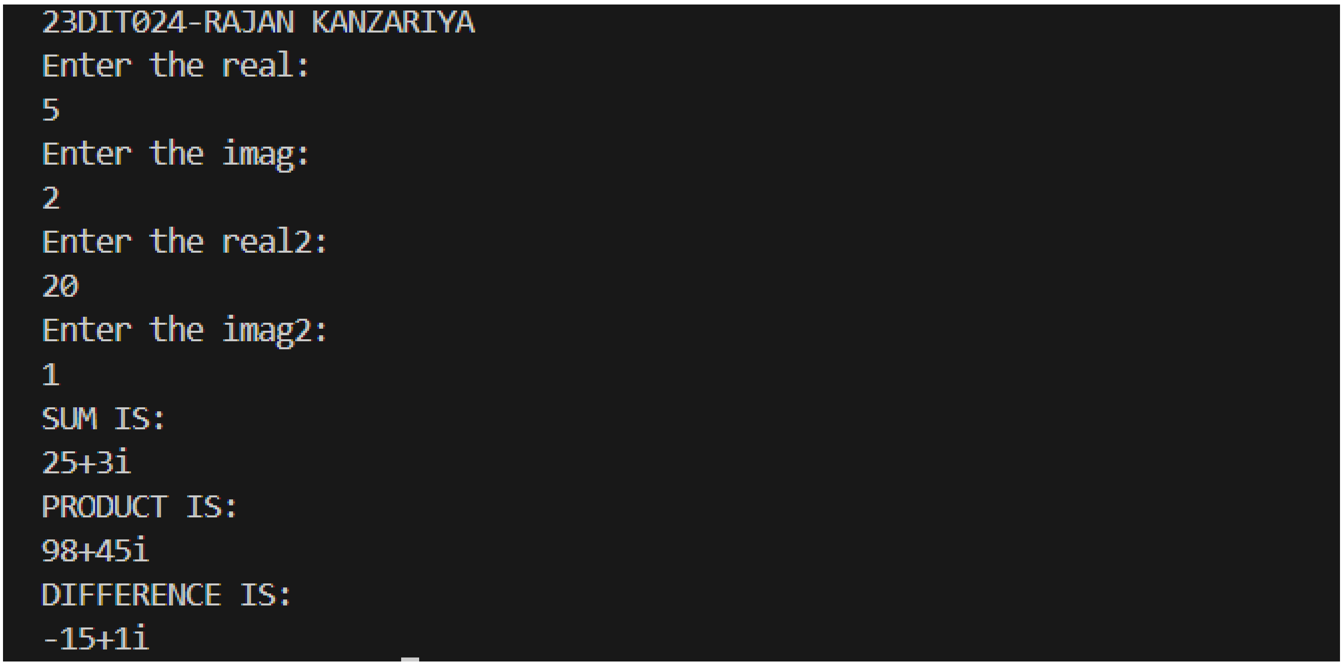
CODE:

```
import java.util.*;

class complex{
    int r1 , i1;

    complex(){
        r1=0;
        i1=0;
    }
    complex(int r, int i){
        r1=r;
        i1=i;
    }

    complex sum(complex c , complex c1){
        complex temp = new complex();
        temp.r1= c.r1+c1.r1;
        temp.i1= c.i1+c1.i1;
        return temp;
    }

    void print(){
        System.out.println(r1 +  "+" + i1 + "i");
    }

    complex product(complex c , complex c1){
        complex temp = new complex();
        temp.r1= (c.r1*c1.r1)-(c.i1*c1.i1);
        temp.i1= (c.r1*c1.i1)+(c.i1*c1.r1);
        return temp;
    }

    complex diff(complex c , complex c1){
        complex temp = new complex();
        temp.r1= c.r1-c1.r1;
        temp.i1= c.i1-c1.i1;
        return temp;
    }
}
```



```
public class prac15 {
    public static void main(String[] args) {
     System.out.println("23DIT024-RAJAN KANZARIYA ");
        int real ,imag , real1, imag1;
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter the real: ");
        real = obj.nextInt();
        System.out.println("Enter the imag: ");
        imag = obj.nextInt();

        System.out.println("Enter the real2: ");
        real1 = obj.nextInt();
        System.out.println("Enter the imag2: ");
        imag1 = obj.nextInt();

        complex c1= new complex(real,imag);
        complex c2= new complex(real1,imag1);
        complex c3= new complex();
        complex c4= new complex();
        c4=c3.sum(c1, c2);
        System.out.println("SUM IS: ");
        c4.print();
        c4=c3.product(c1, c2);
        System.out.println("PRODUCT IS: ");
        c4.print();
        c4=c3.diff(c1, c2);
        System.out.println("DIFFERENCE IS: ");
        c4.print();
    }
}
```

OUTPUT:



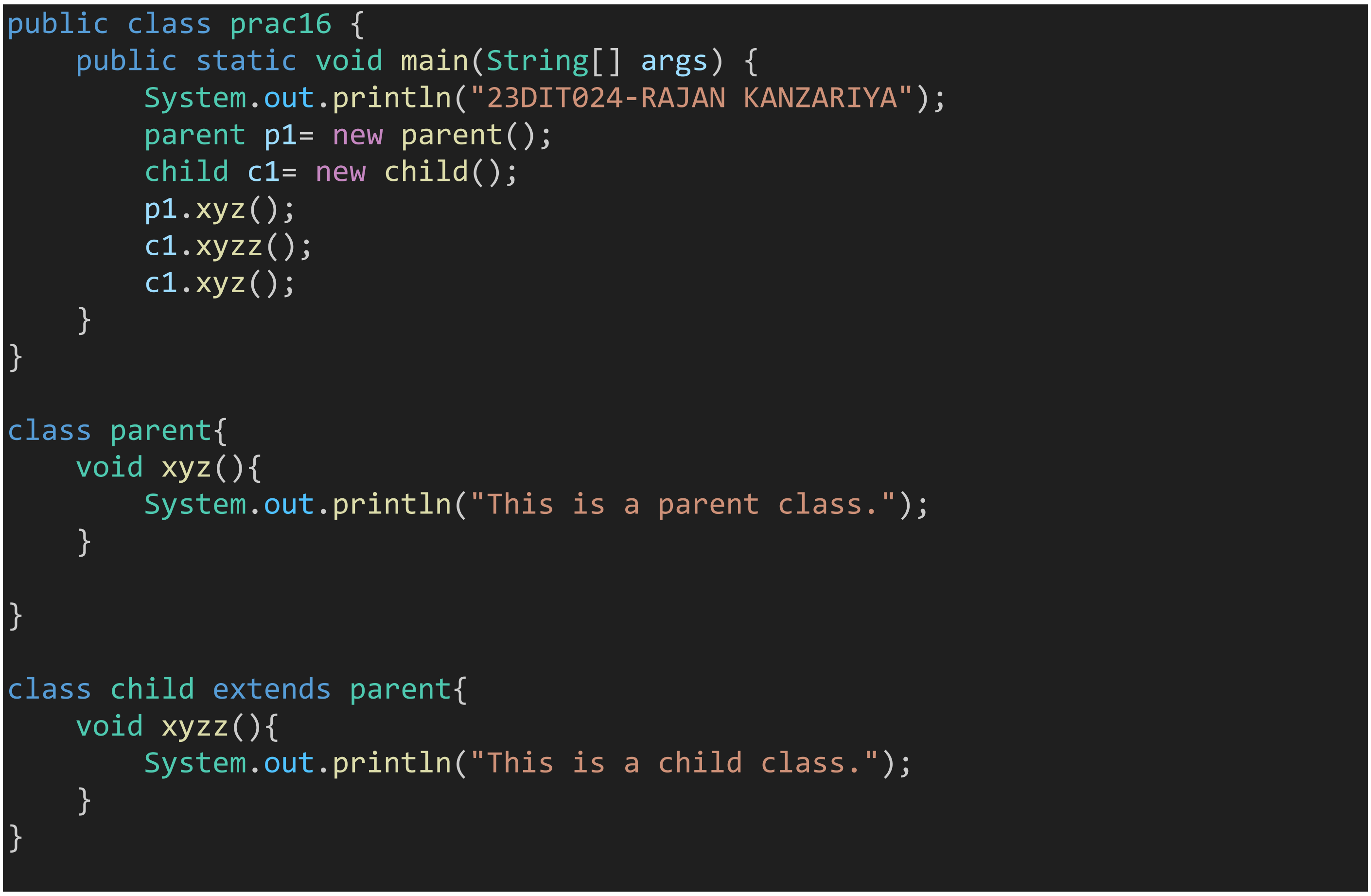
```
23DIT024-RAJAN KANZARIYA
Enter the real:
5
Enter the imag:
2
Enter the real2:
20
Enter the imag2:
1
SUM IS:
25+3i
PRODUCT IS:
98+45i
DIFFERENCE IS:
-15+1i
```

CONCLUSION:

From this practical, I've learned how to develop a class to represent and perform operations on complex numbers. I practiced implementing methods for addition, subtraction, and multiplication of complex numbers. This helped me understand advanced object-oriented programming concepts.

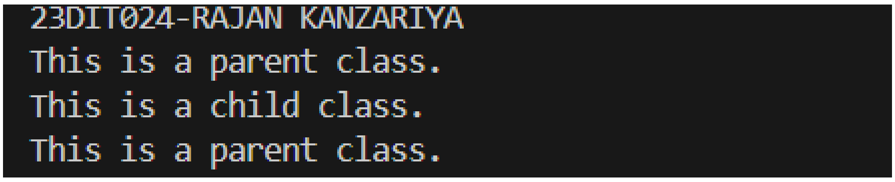
16) AIM : Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call 1 - method of parent class by object of parent class 2 - method of child class by object of child class 3 - method of parent class by object of child class.

CODE:



```java
public class prac16 {
    public static void main(String[] args) {
        System.out.println("23DIT024-RAJAN KANZARIYA");
        parent p1= new parent();
        child c1= new child();
        p1.xyz();
        c1.xyzz();
        c1.xyz();
    }
}

class parent{
    void xyz(){
        System.out.println("This is a parent class.");
    }

}

class child extends parent{
    void xyzz(){
        System.out.println("This is a child class.");
    }
}
```

OUTPUT :



```
23DIT024-RAJAN KANZARIYA
This is a parent class.
This is a child class.
This is a parent class.
```

CONCLUSION :

From this practical, I've learned how to use inheritance by creating parent and child classes with specific methods. I practiced calling methods using objects of different classes. This exercise demonstrated the power of inheritance and method overriding in Java.