1. Write any ten features of java.

Ans :

Here are 10 key features of Java that make it a powerful and widely used programming language:

1. Simple
   - Java is easy to learn and use.
   - Its syntax is clean and concise, and it removes complex features like pointers and operator overloading found in C/C++.
2. Object-Oriented
   - Everything in Java is treated as an object (except primitive types).
   - Concepts like inheritance, encapsulation, abstraction, and polymorphism are core to Java.
3. Platform Independent
   - Java code is compiled into bytecode by the Java Compiler.
   - This bytecode can run on any device that has the Java Virtual Machine (JVM) — making Java "write once, run anywhere".
4. Secure
   - Java has built-in security features like:
     - Bytecode verification
     - Runtime security checks
     - No direct memory access
   - Java applets run inside a sandbox to prevent unauthorized access.
5. Robust
   - Java emphasizes early error checking and runtime exception handling.
   - Features like garbage collection and strong memory management make it less prone to crashes.
6. Multithreaded
   - Java supports multithreading (executing multiple threads concurrently).
   - Useful for developing high-performance applications like games, web servers, etc.
7. Distributed

- Java has networking capabilities using APIs like java.net.
- It can be used to create distributed applications (e.g., using RMI and CORBA).

8. High Performance
   - Although not as fast as C/C++, Java's performance is high due to:
     - Just-In-Time (JIT) compiler
     - Efficient memory management
     - Optimized runtime execution

9. Portable
   - Java bytecode is platform-independent and can run on any system with a compatible JVM.
   - It doesn't depend on hardware-specific features.

10. Dynamic
    - Java supports dynamic loading of classes.
    - It allows programs to load classes at runtime, making it adaptable to evolving environments.

2. What is Unicode and Byte Code?

Ans :

Unicode:

Unicode is a universal character encoding standard that allows the representation of almost all characters from all writing systems (e.g., English, Chinese, Hindi, Arabic, emojis, etc.) used around the world.

In Java:

- Java uses Unicode to internally represent characters (data type char).
- Java's char data type is 16-bit, and it can store any Unicode character.
- This allows Java programs to be internationalized, i.e., support multiple languages and symbols.

Bytecode

Bytecode is an intermediate representation of your Java program. It is the compiled form of Java source code that can be executed by the Java Virtual Machine (JVM).

Key Points:

- When you compile a .java file using the Java compiler (javac), it generates a .class file that contains bytecode.
- Bytecode is platform-independent, which means the same .class file can run on any platform (Windows, Linux, Mac) that has a JVM.
- Bytecode is not human-readable but is machine-readable by the JVM.

3. How java is platform independent?

Ans :

Java achieves platform independence through the use of the **Java Virtual Machine (JVM)**. Here's how the process works:

Compilation Process

1. Write Java Code
   You write your code in .java files.
2. Compile Java Code
   The Java Compiler (javac) compiles .java files into bytecode, which is stored in .class files.
   - Bytecode is not machine-specific, it's a universal, intermediate code.
3. Run with JVM
   The bytecode is executed by the Java Virtual Machine (JVM).
   The JVM translates bytecode into machine code for the underlying operating system.

4. What is JDK and JRE? Differentiate?

Ans :

JDK (Java Development Kit)

- Definition: A software development kit used to develop Java applications.
- Includes:
  - JRE (Java Runtime Environment)
  - Java Compiler (javac)
  - Debuggers, documentation tools (javadoc), and other development tools
- Use: Required when you want to write, compile, and debug Java programs.

◇ JRE (Java Runtime Environment)

- Definition: A software package that provides the environment to run Java programs.
- Includes:
  - JVM (Java Virtual Machine)
  - Core Java libraries
  - Supporting files for execution
- Use: Required only to run Java applications. You cannot develop Java programs with just the JRE.

Feature	JDK (Java Development Kit)	JRE (Java Runtime Environment)
Purpose	To develop and run Java programs	To run Java programs only
Includes	JRE + development tools (e.g. compiler)	JVM + core libraries
Contains JVM?	Yes	Yes
Compiler (javac)	Included	Not Included
Development tools	Included	Not Included
Used by	Java developers	End users who run Java apps

5. What is the task of Class Loader, Verifier, JIT Compiler in JRE.

Ans :

1. Class Loader

Task: Loads Java .class files (bytecode) into memory for execution by the JVM.

Responsibilities:

- Loads classes dynamically at runtime (only when needed).
- Divides loading responsibility into three class loaders:
    1. Bootstrap ClassLoader – loads core Java classes (java.lang, java.util etc.)
    2. Extension ClassLoader – loads classes from ext directory.
    3. Application ClassLoader – loads classes from your application's classpath.
- Ensures that each class is loaded only once into the JVM.
- Supports dynamic loading: classes can be loaded from various sources (disk, network, etc.).

2. Bytecode Verifier (Verifier)

Task: Checks the loaded bytecode for security and integrity before it is executed.

Responsibilities:

- Ensures the bytecode:
    - Follows Java language rules.
    - Does not perform illegal memory access.
    - Does not corrupt the JVM.
- Prevents:
    - Stack overflows/underflows.
    - Access to private/protected data.
    - Code that violates access rights or performs dangerous operations.
- Guarantees the code is safe to execute.

Acts as a security gatekeeper before the code runs.

◇ 3. JIT Compiler (Just-In-Time Compiler)

Task: **Compiles** bytecode into **native machine code at runtime** for better performance.

Responsibilities:

- Converts frequently-used bytecode into **native code** for faster execution.
- Uses **runtime optimizations**:
  - Method inlining
  - Loop unrolling
  - Dead code elimination
- Works alongside the **interpreter**, but gradually **takes over execution** of hot (frequently run) code.
- **Improves speed** by avoiding repeated interpretation of the same bytecode.

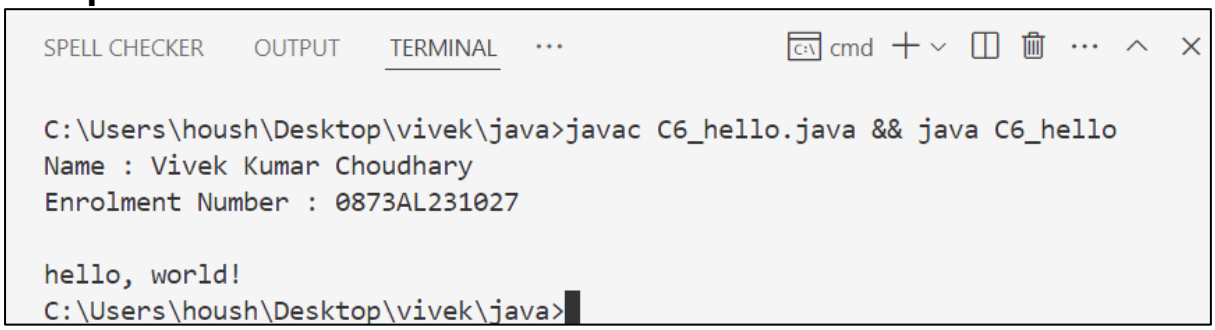
Translates "hot spots" of code into **optimized machine instructions**.

6. Write/compile/execute Hello World program in Java

Ans :

```
public class C6_hello {
    public static void main(String[] args) {
        Name.info();          //method to print name and enrollment
number.
        System.out.print("hello, world!");
    }
}
```

#output



7. Program to find area and circumference of circle

Ans :

```
import java.util.Scanner;
public  class C7_area {
    public static void main(String[] args){
        Name.info();              //method to print name and
enrollment number.
        double radius,area,circumference;
        Scanner scr = new Scanner(System.in);
        System.out.print("enter radius of circle : ");
        radius = scr.nextFloat();
        circumference = 2*3.14*radius;
        area = 3.14*radius*radius;
        System.out.println("area of circle : " +area);
        System.out.println("circumference of circle : "
+circumference);

    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshu\Desktop\vivek\java>javac C7_area.java && java C7_area
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

enter radius of circle : 5
area of circle : 78.5
circumference of circle : 31.400000000000002
```

8. Program to calculate sum of 5 subjects & find percentage.

Ans :

```
import java.util.Scanner;
public class C8_percentage {
    public static void main(String[] args) {
        Scanner scr = new Scanner(System.in);
        Name.info();              //method to print name and
enrollment number.

        double sub1,sub2,sub3,sub4,sub5,result;
        System.out.print("enter marks of subject 1 : ");
        sub1 = scr.nextFloat();

        System.out.print("enter marks of subject 2 : ");
        sub2 = scr.nextFloat();

        System.out.print("enter marks of subject 3 : ");
        sub3 = scr.nextFloat();

        System.out.print("enter marks of subject 4 : ");
        sub4 = scr.nextFloat();

        System.out.print("enter marks of subject 5 : ");
        sub5 = scr.nextFloat();

        result = (sub1+sub2+sub3+sub4+sub5)/5;
        System.out.println("you got " +result+ " %");
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshu\Desktop\vivek\java>javac C8_percentage.java && java C8_percen
tage
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

enter marks of subject 1 : 90
enter marks of subject 2 : 99
enter marks of subject 3 : 95
enter marks of subject 4 : 98
enter marks of subject 5 : 96
you got 95.6 %
```

9. Write a program to calculate simple interest.

Ans :

```
import java.util.Scanner;
public class C9_smp_Interest {
    public static void main(String[] args) {
        Scanner scr = new Scanner(System.in);
        Name.info();              //method to print name and
enrollment number.
        float P ,R,T ,SI;
        System.out.print("enter principle amount : ");
        P = scr.nextFloat();
        System.out.print("enter rate of interest : ");
        R = scr.nextFloat();
        System.out.print("enter time in year : ");
        T = scr.nextFloat();

        SI = (P*R*T)/100;
        System.out.println("simple interest is "+SI);

    }
}
```

#output

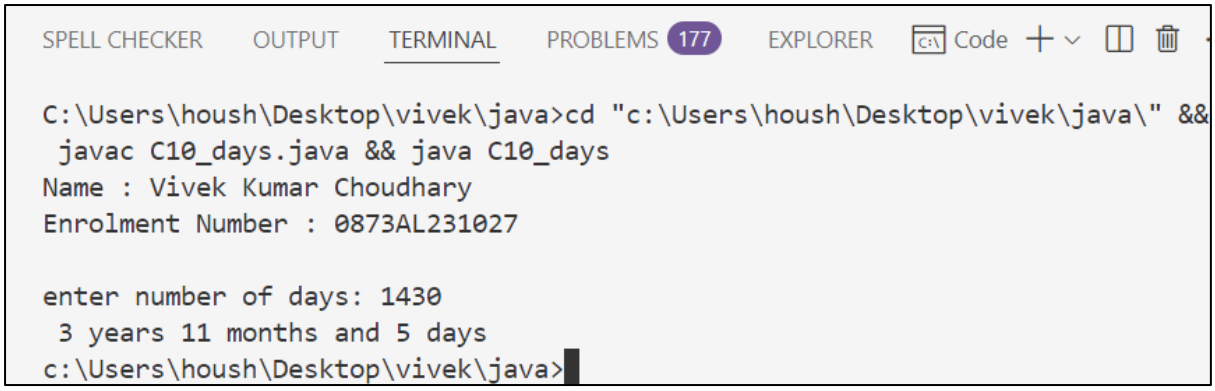
```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

C:\Users\houshu\Desktop\vivek\java>cd "c:\Users\houshu\Desktop\vivek\java\" &&
ivek\java\" && javac C9_smp_Interest.java && java C9_smp_Interest
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

enter principle amount : 100000
enter rate of interest : 5
enter time in year : 2
simple interest is 10000.0
```

10. Write a program which accepts days(eg. 670 days) as integer and display total number of years, months and days in it.

Ans :

```
//days to year, months and days
import java.util.Scanner;
public class C10_days {
    public static void main(String[] args) {
        Name.info();              //method to print name and
enrollment number.
        Scanner scr = new Scanner(System.in);
        int days,years,days_left,month;
        System.out.print("enter number of days: ");
        days = scr.nextInt();
        years = days/365;
        days_left = days%365;
        month = days_left/30;
        days = days_left%30;
        System.out.printf(" %d years %d months and %d
days",years,month,days);
    }
}
```

#output



11. Program to convert temperature from Fahrenheit to Celsius as C= 5*(f-32)/9

Ans :

```
//convert temperature from Fahrenheit to celsius
import java.util.Scanner;
public class C11_temperature {
    public static void main(String[] args) {
        Scanner scr = new Scanner(System.in);
        Name.info();          //method to print name and enrollment
number.
        float Fahrenheit;
        System.out.print("Enter temperature in Fahrenheit : ");
        Fahrenheit = scr.nextFloat();
        System.out.println("temperature in celsius is : " +
(5*(Fahrenheit-32)/9));
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshl\Desktop\vivek\java>cd "c:\Users\houshl\Desktop\vivek\java\" &&
 javac C11_temperature.java && java C11_temperature
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Enter temperature in Fahrenheit : 100
temperature in celsius is : 37.77778
```

12. Program to swap two no's without using third variable.

Ans :

```
import java.util.Scanner;

public class C12_swap {
    public static void main(String[] args){
        Scanner scr = new Scanner(System.in);
        Name n = new Name();
        Name.info();              //method to print name and
enrollment number.
        int num1,num2;
        System.out.print("Enter num1 : ");
        num1 = scr.nextInt();
        System.out.print("Enter num2 : ");
        num2 = scr.nextInt();

        num1 = num1+num2;
        num2 = num1-num2;
        num1 = num1-num2;
        System.out.printf("num1 = %d \nnum2 = %d",num1,num2);
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshu\Desktop\vivek\java>javac C12_swap.java && java C12_swap
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Enter num1 : 14
Enter num2 : 30
num1 = 30
 num2 = 14
```

13. Find the result of following (accept values for variables used in right side of expression)

a. y=x 2 +3x-7 (print value of Z)

b. y=x++ + ++x (print value of x and y)

c. z= x++ - --y - --x + x++ (print value of x ,y and z)

d. z = x && y || !(x||y) (print value of Z)

Ans :

```
import java.util.Scanner;

public class ExpressionEvaluation {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        Scanner scanner = new Scanner(System.in);

        // For (a), (b), (c) assume x and y are integers
        // For (d), assume x and y are booleans

        // a. y = x^2 + 3x - 7
        System.out.print("Enter integer value for x (for a): ");
        int x_a = scanner.nextInt();
        int y_a = x_a * x_a + 3 * x_a - 7;
        System.out.println("a) Value of y = " + y_a);

        // b. y = x++ + ++x
        System.out.print("\nEnter integer value for x (for b): ");
        int x_b = scanner.nextInt();
        int y_b = x_b++ + ++x_b;
        System.out.println("b) After evaluation:");
        System.out.println("   x = " + x_b);
        System.out.println("   y = " + y_b);

        // c. z = x++ - --y - --x + x++
        System.out.print("\nEnter integer value for x (for c): ");
        int x_c = scanner.nextInt();
        System.out.print("Enter integer value for y (for c): ");
        int y_c = scanner.nextInt();
        int z_c = x_c++ - --y_c - --x_c + x_c++;
        System.out.println("c) After evaluation:");
        System.out.println("   x = " + x_c);
        System.out.println("   y = " + y_c);
        System.out.println("   z = " + z_c);

        // d. z = x && y || !(x || y)
        // Here x and y are boolean variables
```

```
        System.out.print("\nEnter boolean value for x (true/false)
(for d): ");
        boolean x_d = scanner.nextBoolean();
        System.out.print("Enter boolean value for y (true/false)
(for d): ");
        boolean y_d = scanner.nextBoolean();
        boolean z_d = (x_d && y_d) || !(x_d || y_d);
        System.out.println("d) Value of z = " + z_d);

        scanner.close();
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshl\Desktop\vivek\java>javac C13.java && java C13
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Enter integer value for x (for a): 14
a) Value of y = 231

Enter integer value for x (for b): 30
b) After evaluation:
   x = 32
   y = 62

Enter integer value for x (for c): 10
Enter integer value for y (for c): 6
c) After evaluation:
   x = 11
   y = 5
   z = 5

Enter boolean value for x (true/false) (for d): true
Enter boolean value for y (true/false) (for d): true
d) Value of z = true
```

14. Program to reverse a given number.

Ans :

```
import java.util.Scanner;
public class C14_reverse
{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        Name.info();              //method to print name and
enrollment number.
        int n,t,rv=0;
        System.out.print("enter number to reverse: ");
        n = sc.nextInt();
        while (n!=0){
            t = n%10;
            rv = (rv*10)+t;
            n = n/10;
        }
        System.out.println("reverse value : " +rv);
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshu\Desktop\vivek\java>javac C14_reverse.java && java C14_reverse

Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

enter number to reverse: 3014
reverse value : 4103
c:\Users\houshu\Desktop\vivek\java>
```

15. In a company an employee is paid as under:
If his basic salary is less than Rs. 1500, then HRA = 10% of basic salary and DA = 90% of basic salary.
If his salary is either equal to or above Rs. 1500, then HRA = Rs. 500 and DA = 98% of basic salary.
If the employees salary is input by the user write a program to find his gross salary. GS=Basic+DA+HRA

Ans :

```
import java.util.Scanner;
public class C15_salary {
    public static void main(String[] args) {
        float salary,gross_salary,DA,HRA,Basic;
        Scanner scr = new Scanner(System.in);
        Name.info();              //method to print name and
enrollment number.
        System.out.print("what's your salary : ");
        salary = scr.nextFloat();
        if (salary > 0 && salary < 1500){
            Basic = salary;
            DA = (90*salary)/100;
            HRA = (10*salary)/100;
            gross_salary = Basic + DA + HRA;
            System.out.println("your gross salary is :
"+gross_salary);
        }else if(salary >= 1500){
            Basic = salary;
            DA = (98*salary)/100;
            HRA = 500;
            gross_salary = Basic + DA + HRA;
            System.out.println("your gross salary is :
"+gross_salary);
        }else{
            System.out.println("wrong input");}}}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshu\Desktop\vivek\java>javac C15_salary.java && java C15_salary
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

what's your salary : 10000000
your gross salary is : 1.98005E7

c:\Users\houshu\Desktop\vivek\java>
```

16. Program to find greatest in 3 numbers.

Ans :

```
import java.util.Scanner;
public class C16_biggest {
    public static void main(String[] args) {
        Scanner scr = new Scanner(System.in);
        Name.info();              //method to print name and
enrollment number.
        int num1,num2,num3,d;
        System.out.print("Enter number 1 : ");
        num1 = scr.nextInt();
        System.out.print("Enter number 2 : ");
        num2 = scr.nextInt();
        System.out.print("Enter number 3 : ");
        num3 = scr.nextInt();
        d = (num1>num2 && num1 > num3)?num1:(num2>num3)?num2:num3;
        System.out.println("greatest number is " +d);

    }

}
```

#output

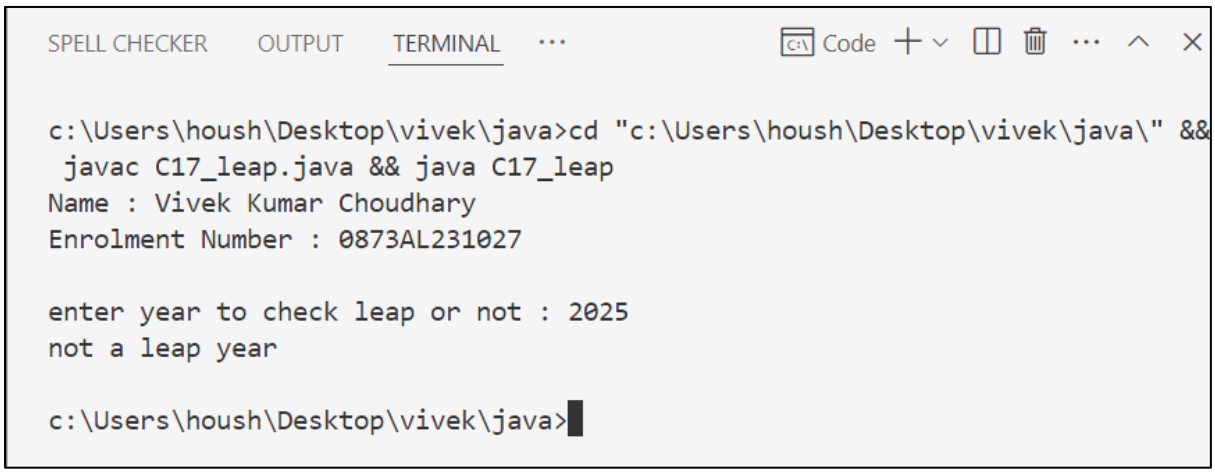
```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshu\Desktop\vivek\java>javac C16_biggest.java && java C16_biggest

Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Enter number 1 : 14
Enter number 2 : 30
Enter number 3 : 95
greatest number is 95
```

17. Program to check that entered year is leap year or not.

Ans :

```
import java.util.Scanner;
public class C17_leap {
    public static void main(String[] args){
        Scanner scr = new Scanner(System.in);
        Name.info();              //method to print name and
enrollment number.
        int year;
        System.out.print("enter year to check leap or not : ");
        year = scr.nextInt();
        if ((year % 400 == 0) || (year%4 == 0 && year%100 !=0)){
                System.out.println("leap year");
        }else{
            System.out.println("not a leap year");
        }
    }
}
```

#output



18. Accept person age(int), gender(int 1 for male and 0 for female), then check whether person is eligible for marriage or not.

Ans :

```
    import java.util.Scanner;
    public class Assi18 {
        public static void main(String[] args) {
            Scanner scr = new Scanner(System.in);
            Name.info();        //method to print name and
enrollment number.
            int age,gender;
            System.out.print("enter your age : ");
            age = scr.nextInt();
            System.out.print("your gender(1 for male & 0 for female)
: ");
            gender = scr.nextInt();
            if (gender == 0){
                if (age > 18){
                    System.out.println("eligible for marriage.");
                }else{
                    System.out.println("not eligible for
marriage.");
                }
            }else if (gender == 1){
                if (age > 21){
                    System.out.println("eligible for marriage.");
                }else{
                    System.out.println("not eligible for
marriage.");
                }
            }else{
                System.out.println("invalid inputs.");
            }}}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshub\Desktop\vivek\java>javac C18_eligible.java && java C18_eligib
le
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

enter your age : 21
your gender(1 for male & 0 for female) : 1
eligible for marriage.

c:\Users\houshub\Desktop\vivek\java>
```

19. Program to print a table of any number.

Ans:

```
import java.util.Scanner;

public class C19_table {
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        Name.info();              //method to print name and
enrollment number.
        int num;
        System.out.print("Enter number to print table : ");
        num = sc.nextInt();
        for(int i = 1;i<=10;i++){
            int result = i*num;
            System.out.printf("%d X %d = %d\n",num,i,result);
        }
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshl\Desktop\vivek\java>javac C19_table.java && java C19_table
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Enter number to print table : 14
14 X 1 = 14
14 X 2 = 28
14 X 3 = 42
14 X 4 = 56
14 X 5 = 70
14 X 6 = 84
14 X 7 = 98
14 X 8 = 112
14 X 9 = 126
14 X 10 = 140

c:\Users\houshl\Desktop\vivek\java>
```

20. Program to check whether number is prime or not

Ans :

```
import java.util.Scanner;

public class C20_prime {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Name.info();          //method to print name and enrollment
number.
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        boolean isPrime = true;
        // 0 and 1 are not prime numbers
        if (num <= 1) {
            isPrime = false;
        } else {
            // Check from 2 to sqrt(num)
            for (int i = 2; i <= Math.sqrt(num); i++) {
                if (num % i == 0) {
                    isPrime = false;
                    break;
                }
            }
        }
        if (isPrime) {
            System.out.println(num + " is a Prime Number.");
        } else {
            System.out.println(num + " is Not a Prime Number.");
        }
        scanner.close();
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshu\Desktop\vivek\java>cd "c:\Users\houshu\Desktop\vivek\java\" &&
 javac C20_prime.java && java C20_prime
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Enter a number: 73
73 is a Prime Number.

c:\Users\houshu\Desktop\vivek\java>
```

21. Calculate series : $1^2 + 2^2 + 3^2 + 4^2 + \ldots\ldots + n^2$

Ans :

```
import java.util.Scanner;
import java.math.*;
public class Assi22 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Name.info();          //method to print name and enrollment
number.
        int num, result = 0;
        System.out.print("enter number : ");
        num = sc.nextInt();
        for(int i = 1;i<=num;i++){
        result += Math.pow(i, 2);
        }

        System.out.printf("sum of series upto %d elements
is   "+result,num);
    }
}
```

#output

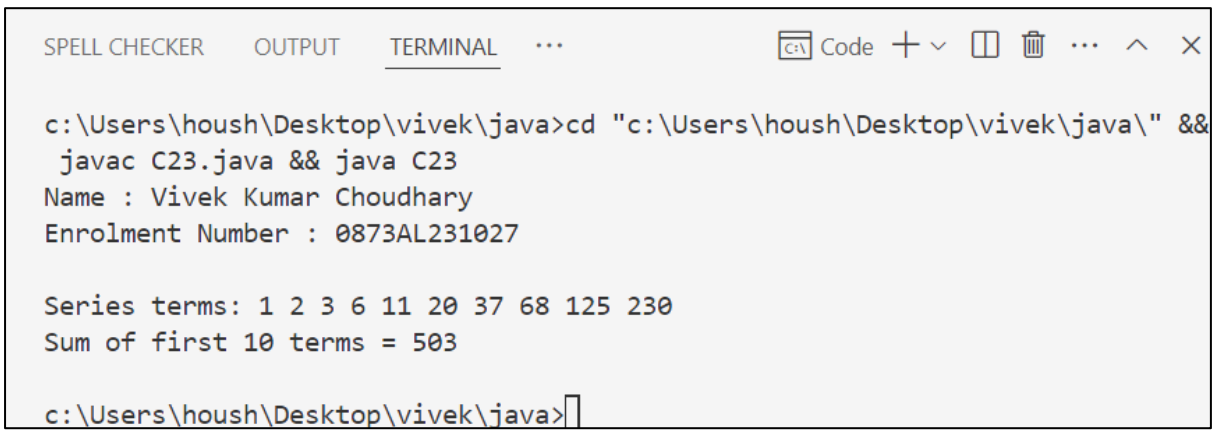
```
SPELL CHECKER 1   OUTPUT   TERMINAL   ···                Code

c:\Users\houshl\Desktop\vivek\java>cd "c:\Users\houshl\Desktop\vivek\java\" &&
 javac C22_series.java && java C22_series
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

enter number : 5
sum of series upto 5 elements is   55
c:\Users\houshl\Desktop\vivek\java>
```

22. Calculate sum of Lucas series (up to 10 terms) : 1, 2, 3, 6, 11, 20 ,.....

Ans :

```
public class C23 {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        int n = 10; // Number of terms
        int[] series = new int[n];

        // Initialize first three terms
        series[0] = 1;
        series[1] = 2;
        series[2] = 3;
        int sum = series[0] + series[1] + series[2];

        // Calculate next terms using series[n] = series[n-1] +
series[n-2] + series[n-3]
        for (int i = 3; i < n; i++) {
            series[i] = series[i - 1] + series[i - 2] + series[i -
3];
            sum += series[i];
        }

        // Print the series
        System.out.print("Series terms: ");
        for (int i = 0; i < n; i++) {
            System.out.print(series[i] + " ");
        }
        System.out.println("\nSum of first " + n + " terms = " +
sum);
    }
}
```

#output



23. Print all prime numbers between two given numbers.

Ans :

```
import java.util.Scanner;
public class C24_prime {
    // Function to check if a number is prime
    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;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Name.info();        //method to print name and enrollment
number.
        // Input range from user
        System.out.print("Enter the starting number: ");
        int start = scanner.nextInt();
        System.out.print("Enter the ending number: ");
        int end = scanner.nextInt();
        System.out.println("Prime numbers between " + start + " and
" + end + " are:");
        // Loop through the range and print primes
        for (int i = start; i <= end; i++) {
            if (isPrime(i)) {
                System.out.print(i + " ");
            }
        }
        scanner.close();
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ...                    Code

c:\Users\housh\Desktop\vivek\java>cd "c:\Users\housh\Desktop\vivek\java\" &&
 javac C24_prime.java && java C24_prime
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Enter the starting number: 5
Enter the ending number: 99
Prime numbers between 5 and 99 are:
5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
c:\Users\housh\Desktop\vivek\java>
```

24. Program to show sum and average of 10 element array. Accept array elements from user.

Ans :

```
import java.util.Scanner;
public class Assi25 {
    public static void main(String[] args) {
        Name.info();        //method to print name and enrollment
number.
        int a[] = new int[10];
        Scanner sc = new Scanner(System.in);
        int num, i, len=10;
        float avg = 0, sum = 0;
        for (i = 0;i < 10;i++){
            System.out.printf("Enter element %d : ",i+1);
            num = sc.nextInt();
            a[i] = num;
        }
        len = a.length;
        for (i = 0;i < 10 ; i++){
            sum += a[i];
        }
        System.out.println("sum = "+sum);
        System.out.println("average = "+sum/len);
    }
}
```

#output

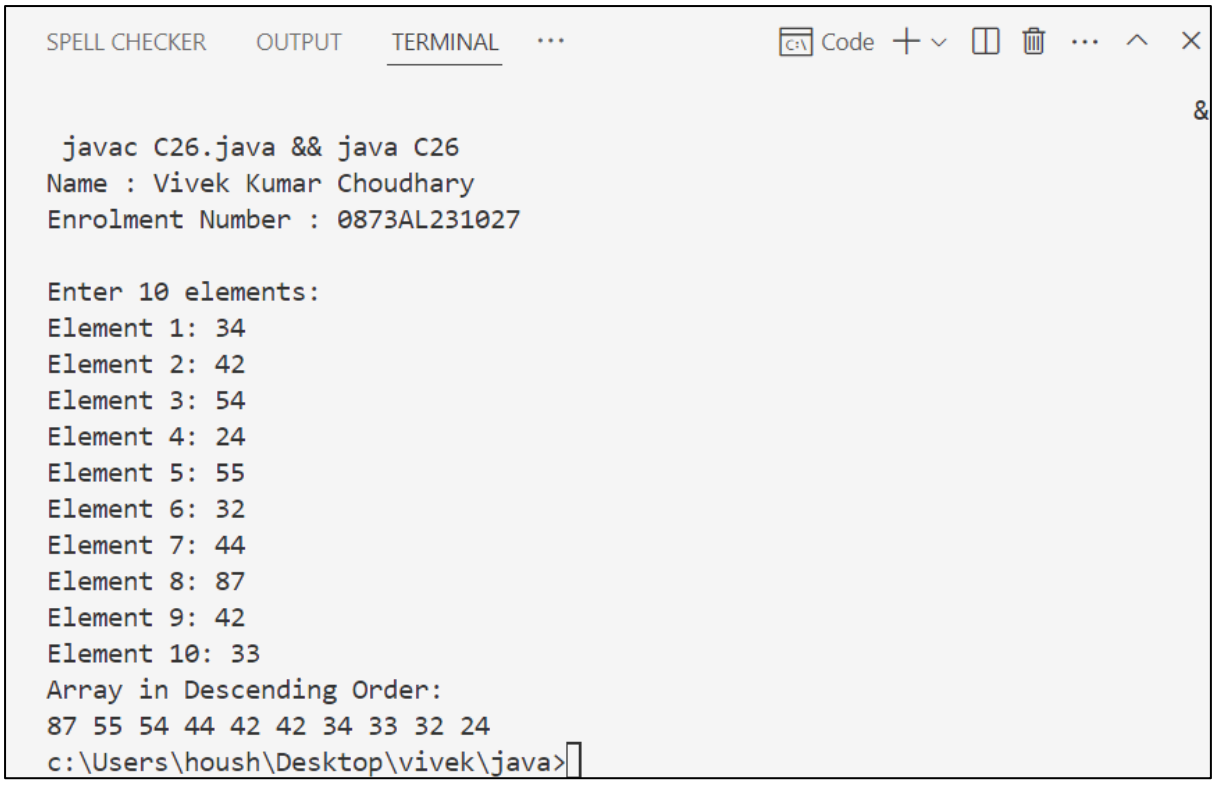
```
SPELL CHECKER 1   OUTPUT   TERMINAL   ···                Code

c:\Users\houshl\Desktop\vivek\java>cd "c:\Users\houshl\Desktop\vivek\java\" &&
 javac C25_sumavg.java && java C25_sumavg
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Enter element 1 : 76
Enter element 2 : 44
Enter element 3 : 33
Enter element 4 : 53
Enter element 5 : 22
Enter element 6 : 74
Enter element 7 : 22
Enter element 8 : 85
Enter element 9 : 25
Enter element 10 : 56
sum = 490.0
average = 49.0

c:\Users\houshl\Desktop\vivek\java>
```

25. Sort ten element array in descending order.

Ans :

```
import java.util.Scanner;
import java.util.Arrays;
public class C26 {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number
        Scanner scanner = new Scanner(System.in);
        int[] arr = new int[10];// Input 10 elements
        System.out.println("Enter 10 elements:");
        for (int i = 0; i < 10; i++) {
            System.out.print("Element " + (i + 1) + ": ");
            arr[i] = scanner.nextInt();
        }// Sort in descending order using simple bubble sort
        for (int i = 0; i < arr.length - 1; i++) {
            for (int j = 0; j < arr.length - i - 1; j++) {
                if (arr[j] < arr[j + 1]) {  // Swap arr[j] and arr[j
+ 1]
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;}}}// Output the sorted array
        System.out.println("Array in Descending Order:");
        for (int num : arr) {
            System.out.print(num + " ");}
        scanner.close();}}
```

#output



26. Create a array of 17 elements in 5 rows. And calculate sum of all elements.

Ans:

```
import java.util.Scanner;

public class C27 {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        Scanner scanner = new Scanner(System.in);

        int rows = 5;
        int cols = 4; // We'll use 4 columns per row, but last row
will have only 1 element (5*4=20 > 17)
        int[][] arr = new int[rows][cols];
        int count = 0, totalElements = 17, sum = 0;

        System.out.println("Enter 17 elements:");

        for (int i = 0; i < rows && count < totalElements; i++) {
            for (int j = 0; j < cols && count < totalElements; j++)
{
                System.out.print("Enter element " + (count + 1) + ":
");
                arr[i][j] = scanner.nextInt();
                sum += arr[i][j];
                count++;
            }
        }

        // Display array in matrix format
        System.out.println("\nArray elements in 5-row layout:");
        count = 0;
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                if (count < totalElements) {
                    System.out.print(arr[i][j] + "\t");
                    count++;
                }
            }
            System.out.println();
        }

        // Output total sum
        System.out.println("\nSum of all 17 elements: " + sum);

        scanner.close();
    }
```

```
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                                Code

c:\Users\houshu\Desktop\vivek\java>javac C27.java && java C27
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Enter 17 elements:
Enter element 1: 45
Enter element 2: 32
Enter element 3: 33
Enter element 4: 44
Enter element 5: 55
Enter element 6: 77
Enter element 7: 43
Enter element 8: 22
Enter element 9: 19
Enter element 10: 49
Enter element 11: 87
Enter element 12: 75
Enter element 13: 57
Enter element 14: 54
Enter element 15: 94
Enter element 16: 87
Enter element 17: 49

Array elements in 5-row layout:
45      32      33      44
55      77      43      22
19      49      87      75
57      54      94      87
49

Sum of all 17 elements: 922

c:\Users\houshu\Desktop\vivek\java>
```

27. Program to find multiplication of two 3X3 matrices.

Ans :

```
import java.util.Scanner;

public class C28 {
    public static void main(String[] args) {
        Name.info(); // Print name and enrollment number

        Scanner scanner = new Scanner(System.in);

        int[][] A = new int[3][3];  // First matrix
        int[][] B = new int[3][3];  // Second matrix
        int[][] result = new int[3][3];  // Resultant matrix

        // Input for first matrix
        System.out.println("Enter elements of first 3x3 matrix
(A):");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print("A[" + i + "][" + j + "] = ");
                A[i][j] = scanner.nextInt();
            }
        }

        // Input for second matrix
        System.out.println("\nEnter elements of second 3x3 matrix
(B):");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print("B[" + i + "][" + j + "] = ");
                B[i][j] = scanner.nextInt();
            }
        }

        // Matrix multiplication logic
        for (int i = 0; i < 3; i++) {          // Rows of A
            for (int j = 0; j < 3; j++) {      // Columns of B
                result[i][j] = 0;
                for (int k = 0; k < 3; k++) { // Columns of A / Rows
of B
                    result[i][j] += A[i][k] * B[k][j];
                }
            }
        }

        // Display Resultant Matrix
        System.out.println("\nResultant Matrix (A × B):");
        for (int i = 0; i < 3; i++) {
```

```
            for (int j = 0; j < 3; j++) {
                System.out.print(result[i][j] + "\t");
            }
            System.out.println();
        }

        scanner.close();
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                                  Code

c:\Users\houshu\Desktop\vivek\java>cd "c:\Users\houshu\Desktop\vivek\java\" &&
 javac C28.java && java C28
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Enter elements of first 3x3 matrix (A):
A[0][0] = 34
A[0][1] = 22
A[0][2] = 44
A[1][0] = 22
A[1][1] = 54
A[1][2] = 00
A[2][0] = 43
A[2][1] = 24
A[2][2] = 32

Enter elements of second 3x3 matrix (B):
B[0][0] = 44
B[0][1] = 22
B[0][2] = 11
B[1][0] = 53
B[1][1] = 00
B[1][2] = 35
B[2][0] = 00
B[2][1] = 35
B[2][2] = 64

Resultant Matrix (A ? B):
2662    2288    3960
3830    484     2132
3164    2066    3361

c:\Users\houshu\Desktop\vivek\java>
```

28. Program to print transpose of a matrix.

Ans :

```
import java.util.Scanner;

public class C29 {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        Scanner scanner = new Scanner(System.in);

        // Get dimensions of matrix
        System.out.print("Enter number of rows: ");
        int rows = scanner.nextInt();

        System.out.print("Enter number of columns: ");
        int cols = scanner.nextInt();

        int[][] matrix = new int[rows][cols];
        int[][] transpose = new int[cols][rows];

        // Input matrix elements
        System.out.println("Enter elements of the matrix:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print("matrix[" + i + "][" + j + "] = ");
                matrix[i][j] = scanner.nextInt();
            }
        }

        // Transpose logic
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                transpose[j][i] = matrix[i][j];
            }
        }

        // Display original matrix
        System.out.println("\nOriginal Matrix:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print(matrix[i][j] + "\t");
            }
            System.out.println();
        }

        // Display transposed matrix
        System.out.println("\nTranspose of Matrix:");
        for (int i = 0; i < cols; i++) {
```

```
            for (int j = 0; j < rows; j++) {
                System.out.print(transpose[i][j] + "\t");
            }
            System.out.println();
        }

        scanner.close();
    }
}
```

#output

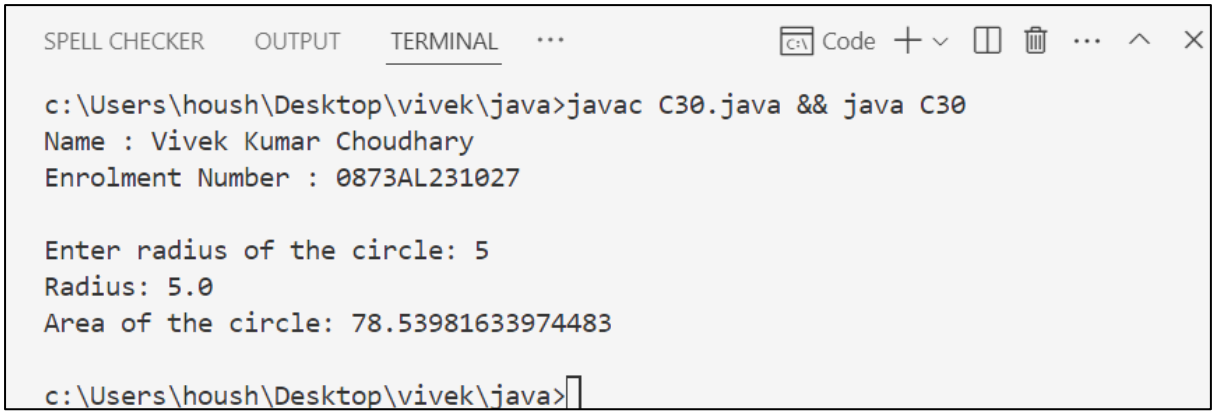
```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshu\Desktop\vivek\java>javac C29.java && java C29
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Enter number of rows: 2
Enter number of columns: 3
Enter elements of the matrix:
matrix[0][0] = 56
matrix[0][1] = 22
matrix[0][2] = 00
matrix[1][0] = 11
matrix[1][1] = 36
matrix[1][2] = 56

Original Matrix:
56      22      0
11      36      56

Transpose of Matrix:
56      11
22      36
0       56

c:\Users\houshu\Desktop\vivek\java>
```

29. Create a class to calculate Area of circle with one data member to store the radius and another to store area value. Create method members
1. init - to input radius from user
2. calc - to calculate area
3. display- to display area

Ans :

```
import java.util.Scanner;

public class C30 {
    double radius;   // Data member to store radius
    double area;     // Data member to store area

    // Method to input radius
    void init() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter radius of the circle: ");
        radius = scanner.nextDouble();
        scanner.close();
    }

    // Method to calculate area
    void calc() {
        area = Math.PI * radius * radius;
    }

    // Method to display area
    void display() {
        System.out.println("Radius: " + radius);
        System.out.println("Area of the circle: " + area);
    }

    // Main method
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        C30 circle = new C30();
        circle.init();     // Input radius
        circle.calc();     // Calculate area
        circle.display();  // Display result
    }
}
```

#output


SPELL CHECKER
OUTPUT
TERMINAL
Code
c:\Users\houshu\Desktop\vivek\java>javac C30.java && java C30
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027
Enter radius of the circle: 5
Radius: 5.0
Area of the circle: 78.53981633974483
c:\Users\houshu\Desktop\vivek\java>

30. Create a class MathOperation with two data member X and Y to store the operand and third data member R to store result of operation.
Create method members
- **init - to input X and Y from user**
- **add - to add X and Y and store in R**
- **multiply - to multiply X and Y and store in R**
- **power - to calculate X Y and store in R**
- **display- to display Result R**

Ans :

```
import java.util.Scanner;

public class C31{
    double X, Y;  // operands
    double R;     // result

    // Method to input X and Y
    void init() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter value for X: ");
        X = scanner.nextDouble();

        System.out.print("Enter value for Y: ");
        Y = scanner.nextDouble();
        scanner.close();
    }

    // Method to add X and Y
    void add() {
        R = X + Y;
    }

    // Method to multiply X and Y
    void multiply() {
        R = X * Y;
    }

    // Method to calculate X^Y
    void power() {
        R = Math.pow(X, Y);
    }

    // Method to display result
    void display() {
        System.out.println("Result (R): " + R);
    }
```

```
    // Main method
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        C31 op = new C31();

        op.init();        // Take input
        op.add();         // Perform addition
        System.out.print("Addition: ");
        op.display();     // Display result

        op.multiply();    // Perform multiplication
        System.out.print("Multiplication: ");
        op.display();     // Display result

        op.power();       // Perform power
        System.out.print("Power (X^Y): ");
        op.display();     // Display result
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshl\Desktop\vivek\java>javac C31_mathOperation.java && java C31_m
athOperation
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Enter value for X: 5
Enter value for Y: 14
Addition: Result (R): 19.0
Multiplication: Result (R): 70.0
Power (X^Y): Result (R): 6.103515625E9

c:\Users\houshl\Desktop\vivek\java>
```

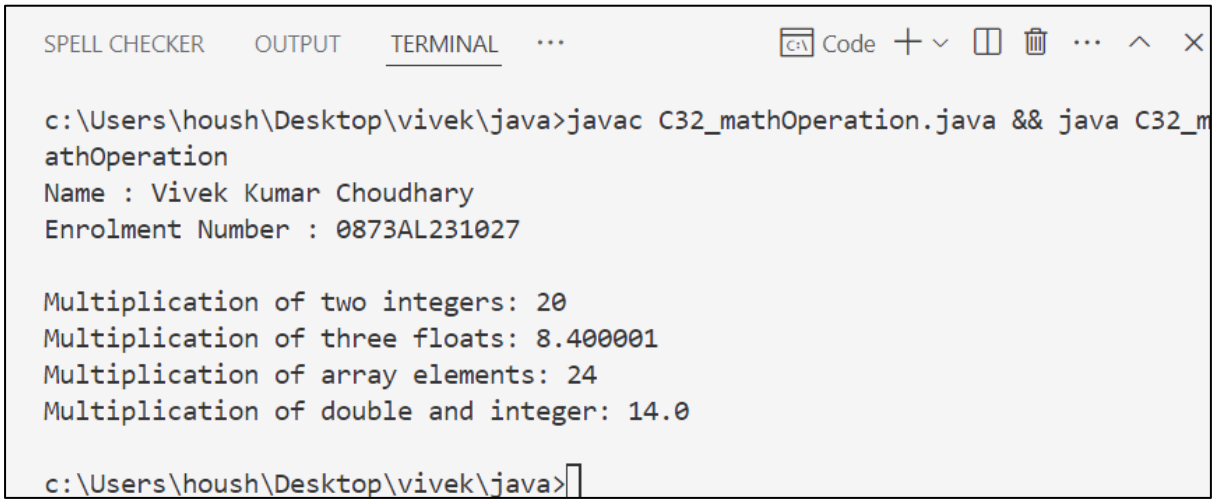
31. Create a class MathOperation containing method 'multiply' to calculate multiplication of following arguments.
a. two integers
b. three float
c. all elements of array
d. one double and one integer

Ans :

```
public class C32_mathOperation{

    // a. Multiply two integers
    void multiply(int a, int b) {
        int result = a * b;
        System.out.println("Multiplication of two integers: " + result);
    }

    // b. Multiply three floats
    void multiply(float a, float b, float c) {
        float result = a * b * c;
        System.out.println("Multiplication of three floats: " + result);
    }

    // c. Multiply all elements of an integer array
    void multiply(int[] arr) {
        int result = 1;
        for (int num : arr) {
            result *= num;
        }
        System.out.println("Multiplication of array elements: " + result);
    }

    // d. Multiply one double and one integer
    void multiply(double a, int b) {
        double result = a * b;
        System.out.println("Multiplication of double and integer: " + result);
    }

    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        C32_mathOperation mo = new C32_mathOperation();

        // Testing all versions of multiply
        mo.multiply(4, 5);                          // two integers
        mo.multiply(1.2f, 3.5f, 2.0f);              // three floats
        mo.multiply(new int[]{2, 3, 4});            // array elements
        mo.multiply(3.5, 4);                       // double and integer
```

```
    }
}
```

#output


SPELL CHECKER
OUTPUT
TERMINAL
Code
c:\Users\houshu\Desktop\vivek\java>javac C32_mathOperation.java && java C32_mathOperation
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027
Multiplication of two integers: 20
Multiplication of three floats: 8.400001
Multiplication of array elements: 24
Multiplication of double and integer: 14.0
c:\Users\houshu\Desktop\vivek\java>

32. What are the two types of Exceptions in Java? What are the differences between them?

Ans :

Types of Exceptions in Java

1. Checked Exceptions (Compile-Time Exceptions)

- Checked at **compile-time** by the compiler.
- Must be **handled using try-catch** or **declared using throws**.
- Typically caused by **external issues** (like missing files, database errors).
- Programs **will not compile** if not handled properly.
- All checked exceptions are **subclasses of Exception** (excluding RuntimeException).

Examples:

- IOException
- FileNotFoundException
- SQLException
- ParseException

2. Unchecked Exceptions (Runtime Exceptions)

- Checked at **runtime**, **not** at compile-time.
- Handling them is **optional**.
- Usually caused by **programming errors** (like null references, divide by zero).
- Program will **compile** but may throw exceptions during execution.
- All unchecked exceptions are **subclasses of RuntimeException**.

Examples:

- NullPointerException
- ArithmeticException
- ArrayIndexOutOfBoundsException
- IllegalArgumentException

Feature	Checked Exception	Unchecked Exception
Checked by compiler	Yes	No
Occurs at	Compile-time	Runtime
Handling required?	Yes (mandatory)	No (optional)
Caused by	External issues (files, DB, network, etc.)	Programming mistakes (logic errors)
Subclass of	Exception (excluding RuntimeException)	RuntimeException
Examples	IOException, SQLException	NullPointerException, ArithmeticException

33. What are the Memory Allocations available in Java?

Ans :

Memory Allocations in Java

Java provides several types of memory areas where different types of data and objects are stored during the execution of a program.

◇ 1. Heap Memory

- Stores: **Objects and class instances**.
- Managed by: **Garbage Collector**.
- Accessible from: **All parts of the application**.
- Grows as needed (up to max heap size).

Example: When you use new to create an object, it goes to the heap.

```
Student s = new Student(); // Stored in heap memory
```

◇ 2. Stack Memory

- Stores: **Method-local variables and function calls**.
- Each thread has its **own stack**.
- Fast access and memory is **automatically deallocated** once method exits.
- **No Garbage Collection** needed.

Example: Primitive types or object references inside methods.

```
int x = 5; // Stored in stack memory
```

◇ 3. Method Area (or Metaspace in Java 8+)

- Stores: **Class metadata**, method data, static variables, constants.
- Shared across all threads.
- In Java 8 and later, Metaspace replaces the PermGen space.

◇ 4. Program Counter (PC) Register

- Each thread has its own **PC Register**.
- Stores the **address of the current instruction** being executed.
- Helps the JVM track **execution flow per thread**.

◇ 5. Native Method Stack

- Used for: **Native method execution (non-Java code)**.
- Stores: **Native (platform-dependent) method calls**.
- Works with JNI (Java Native Interface).

34. Explain final, finally, finalize.

Ans :

➢ `final` – Keyword

Used to declare constants, prevent method overriding, and inheritance.

◇ *Usage of `final`:*

Where Used	Meaning
Variable	Value cannot be changed after initialization (acts as a constant).
Method	Method cannot be overridden in a subclass.
Class	Class cannot be extended (inherited).

Example:

```
final int x = 10;           // x cannot be changed
final class A { }           // No class can extend A
class B {
    final void show() { }   // This method cannot be overridden
}
```

➢ `finally` – Block

Used in **exception handling** to execute code **regardless of exception** occurrence.

◇ *Purpose:*

- Always executes, whether an exception is thrown or not.
- Typically used to **close resources** like files, DB connections, etc.

Example:

```
try {
    int a = 5 / 0; // Throws ArithmeticException
     } catch (ArithmeticException e) {
         System.out.println("Error: " + e);
     } finally {
         System.out.println("This block always runs.");
     }
```

➢ `finalize()` – Method

A method called by **Garbage Collector** before reclaiming an object's memory (deprecated since Java 9+).

◇ *Purpose:*

- Used to **perform cleanup operations** before an object is garbage collected (like closing a file or socket).
- Can be **overridden** in your class.

  Example:

```
protected void finalize() throws Throwable {
    System.out.println("Object is being garbage collected");
}
```

35. What is a singleton class in Java? And How to break the singleton class object?

Ans :

Singleton Class in Java

- A class that allows only **one object (instance)** to be created.
- Provides a **global access point** to that single instance.
- Commonly used for **logging**, **configuration settings**, **thread pools**, etc.

Features of Singleton:

- **Private constructor** to restrict instantiation.
- **Static variable** to hold the single instance.
- **Public static method** to provide access to the instance.

Ways to Break Singleton in Java:

1. Reflection

- Can access private constructor and create a new instance.
- **Prevention**: Throw exception if instance already exists inside constructor.

2. Serialization

- Deserializing creates a new object instance.
- **Prevention**: Implement readResolve() to return existing instance.

3. Cloning

- Object clone creates a copy of the instance.
- **Prevention**: Override clone() and throw CloneNotSupportedException.

4. Multithreading

- Multiple threads might create multiple instances simultaneously.
- **Prevention**: Use synchronized block or volatile keyword for thread-safety.

36. Differentiate between instance and local variables.

Ans :

Feature	Instance Variable	Local Variable
Definition	Variable declared inside a class but **outside any method**	Variable declared **inside a method**, constructor, or block
Scope	Accessible by **all methods** in the class	Accessible **only within the method/block** it is declared
Memory Allocation	Allocated in **heap memory** when object is created	Allocated in **stack memory** during method execution
Default Value	Gets a **default value** (e.g., 0, null, false)	**No default value**, must be explicitly initialized
Access Modifier Allowed	Yes (e.g., private, public, etc.)	Not allowed (declared within methods only)
Existence	Exists **as long as the object exists**	Exists **only while the method is executing**
Use Case	To maintain object **state**	To perform **temporary calculations or processing**

37. Can you call a constructor of a class inside another constructor of same class?

Ans :

Yes, in Java, you can call one constructor from another in the **same class** using this().

Explanation in Bullet Points:

- this() is used to call another constructor of the **same class**.
- It must be the **first line** in the constructor body.
- Helps in **code reuse** and **avoids duplication**.
- Known as **constructor chaining**.
- Useful for providing **default values** or **overloaded constructors**.

38.explain the Java thread lifecycle?

Ans :

A thread in Java goes through **five main states** during its lifecycle, defined in the `Thread.State` enum.

1. New

- Thread is created using Thread class, but not yet started.
- It exists but is not running.
- Example:

```
Thread t = new Thread();
```

2. Runnable

- Thread is started using start() method.
- It's ready to run and waiting for CPU time.
- May move back to Runnable after being paused or waiting.
- Example:

```
t.start();
```

3. Running

- Thread is **actively executing** in CPU.
- Managed by the thread scheduler — **you can't force a thread into this state manually**.
- Only one thread runs per core at a time.

4. Blocked / Waiting / Timed Waiting

- Thread is **temporarily not eligible to run**.
- **Blocked**: Waiting to acquire a lock.
- **Waiting**: Waiting indefinitely for another thread to perform an action.
- **Timed Waiting**: Waits for a specified amount of time.
- Examples:

```
Thread.sleep(1000);      // Timed Waiting
thread.join();           // Waiting
synchronized(lock) { }   // Blocked if another thread holds the lock
```

5. Terminated (Dead)

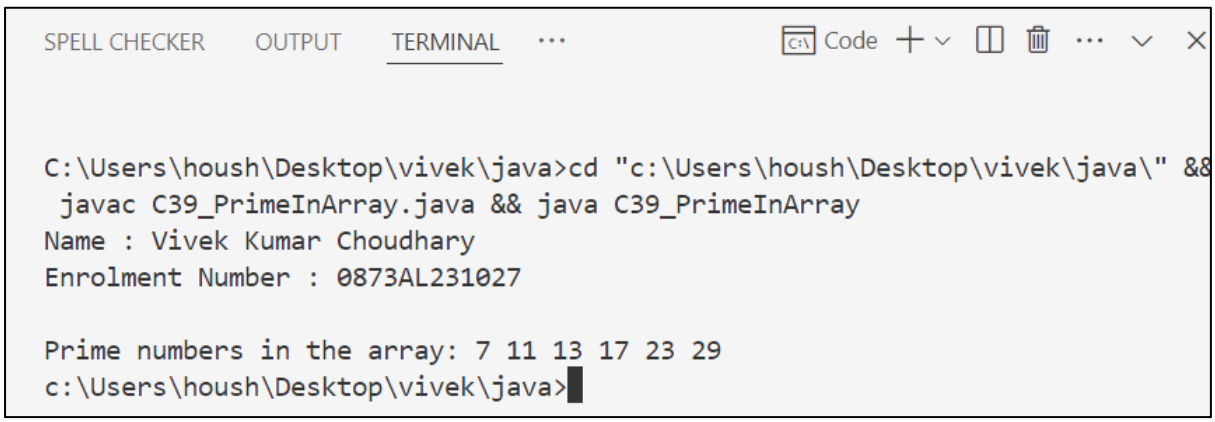
- Thread has finished execution or was forcefully stopped.
- Cannot be restarted again.
- Happens when run() method completes or an uncaught exception occurs.

39. Write a prog. to find prime numbers in an array.

Ans :

```
public class C39_PrimeInArray {
    // Method to check if a number is prime
    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;
    }
    public static void main(String[] args) {
        Name.info();    //to print name and roll number
        int[] arr = {10, 7, 4, 11, 13, 17, 20, 23, 29};

        System.out.print("Prime numbers in the array: ");
        for (int num : arr) {
            if (isPrime(num)) {
                System.out.print(num + " ");
            }
        }
    }
}
```

#output



40. Can we inherit a Constructor?

Ans :

No, constructors cannot be inherited in Java.

Explanation in Bullet Points:

- A constructor is not inherited because it's not a member of the class.
- Constructors are special methods used to initialize objects of a class.
- When a subclass is created, it does not inherit the constructor from the superclass.
- However, a subclass can call the constructor of the superclass using the super() keyword.
- The subclass can also define its own constructors.

41. How will you implement method overloading in Java?

Ans :

Method Overloading in Java means defining multiple methods with the same name in the same class, but with different parameters.

Ways to Implement Method Overloading:

1. By changing the number of parameters
2. By changing the type of parameters
3. By changing the order of parameters (of different types)

42. What is Runtime Polymorphism?

Ans : **Runtime Polymorphism** (also known as **Dynamic Method Dispatch**) is a form of polymorphism where the method call is resolved **at runtime**, not at compile time.

Key Points (Bullet Format):

- Occurs when a **subclass overrides** a method of its **superclass**.
- The **method to be executed is determined at runtime** based on the object type, not the reference type.
- Achieved using **method overriding** and **inheritance**.
- Uses **dynamic binding** (also called **late binding**).
- Enables **Java to support one interface, multiple implementations** (OOP principle).

43. How does Garbage Collection work in Java?

Ans :

Garbage Collection (GC) Overview:

- Java automatically manages memory using **Garbage Collection**.
- GC **identifies and removes objects** that are no longer **reachable** or **needed** by the program.
- This helps prevent **memory leaks** and **optimizes memory usage**.

How Garbage Collection Works:

1. **Object Creation:**
   - When you create objects using new, memory is allocated on the **heap**.
2. **Reachability:**
   - Objects are considered **reachable** if they can be accessed through any chain of references from **root objects** (like local variables, static fields).
3. **Marking:**
   - GC **marks** all reachable objects.
4. **Sweeping:**
   - Objects **not marked** (unreachable) are considered garbage.
   - These objects' memory is **freed** and returned to the heap for future allocation.
5. **Compaction (optional):**
   - Some GC algorithms **compact** memory by moving objects to reduce fragmentation.

Types of Garbage Collectors in Java:

- **Serial Garbage Collector:** Single-threaded, suitable for small apps.
- **Parallel Garbage Collector:** Uses multiple threads for faster GC.
- **CMS (Concurrent Mark Sweep):** Low pause time, good for interactive apps.
- **G1 Garbage Collector:** Divides heap into regions, balances throughput and pause times.

44. Write a Java program to create an ArrayList, add some colors (as strings), and print the collection.

Ans :

```
import java.util.ArrayList;

public class C44_ColorList {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        // Create an ArrayList of Strings
        ArrayList<String> colors = new ArrayList<>();

        // Add some colors to the ArrayList
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");
        colors.add("Yellow");
        colors.add("Orange");

        // Print the ArrayList
        System.out.println("Colors in the list: " + colors);
    }
}
```

#output

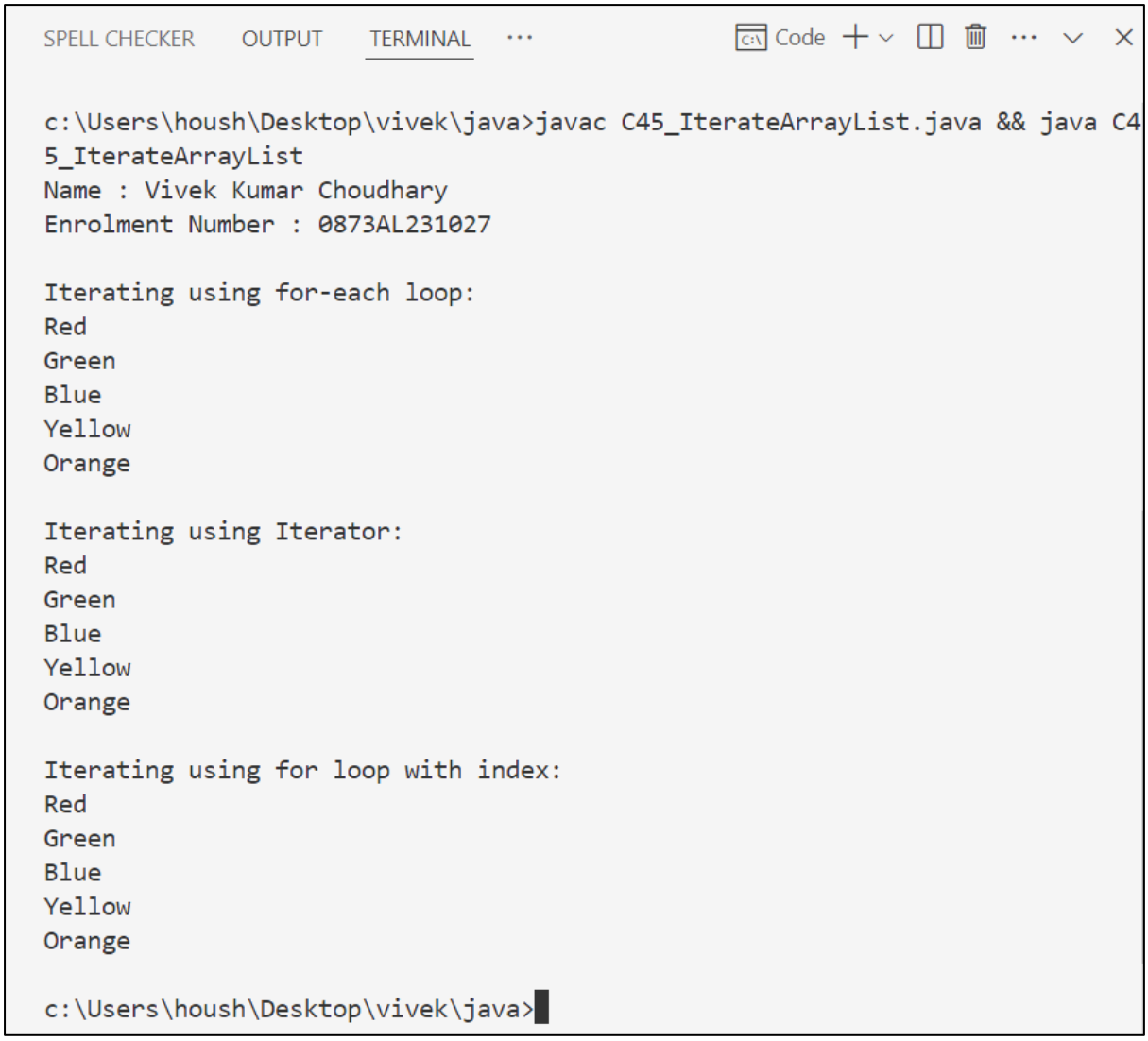
```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshl\Desktop\vivek\java>javac C44_ColorList.java && java C44_Color
List
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Colors in the list: [Red, Green, Blue, Yellow, Orange]

c:\Users\houshl\Desktop\vivek\java>
```

45. Write a Java program to iterate through all elements in an ArrayList.

Ans :

```
import java.util.ArrayList;
import java.util.Iterator;

public class C45_IterateArrayList {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        // Create an ArrayList of Strings
        ArrayList<String> colors = new ArrayList<>();

        // Add some colors
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");
        colors.add("Yellow");
        colors.add("Orange");

        System.out.println("Iterating using for-each loop:");
        for (String color : colors) {
            System.out.println(color);
        }

        System.out.println("\nIterating using Iterator:");
        Iterator<String> it = colors.iterator();
        while (it.hasNext()) {
            System.out.println(it.next());
        }

        System.out.println("\nIterating using for loop with
index:");
        for (int i = 0; i < colors.size(); i++) {
            System.out.println(colors.get(i));
        }
    }
}
```

#output


SPELL CHECKER
OUTPUT
TERMINAL
Code
c:\Users\houshu\Desktop\vivek\java>javac C45_IterateArrayList.java && java C45_IterateArrayList
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027
Iterating using for-each loop:
Red
Green
Blue
Yellow
Orange
Iterating using Iterator:
Red
Green
Blue
Yellow
Orange
Iterating using for loop with index:
Red
Green
Blue
Yellow
Orange
c:\Users\houshu\Desktop\vivek\java>

46. Write a Java program to insert an element into the ArrayList at the first position.

Ans :

```
import java.util.ArrayList;

public class C46_InsertAtFirstPosition {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>();

        // Add some initial colors
        colors.add("Green");
        colors.add("Blue");
        colors.add("Yellow");

        System.out.println("Before insertion: " + colors);

        // Insert an element at the first position (index 0)
        colors.add(0, "Red");

        System.out.println("After insertion at first position: " +
colors);
    }
}
```

#output

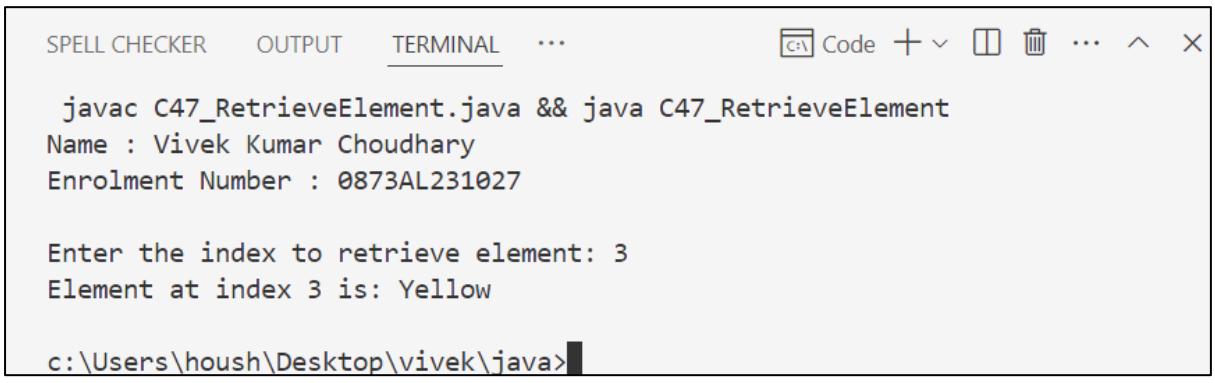
```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

 javac C46_InsertAtFirstPosition.java && java C46_InsertAtFirstPosition
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Before insertion: [Green, Blue, Yellow]
After insertion at first position: [Red, Green, Blue, Yellow]

c:\Users\houshl\Desktop\vivek\java>
```

47. Write a Java program to retrieve an element at a specified index from a given ArrayList.

Ans :

```
import java.util.ArrayList;
import java.util.Scanner;

public class C47_RetrieveElement {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>();
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");
        colors.add("Yellow");
        colors.add("Orange");

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the index to retrieve element: ");
        int index = sc.nextInt();

        if (index >= 0 && index < colors.size()) {
            String element = colors.get(index);
            System.out.println("Element at index " + index + " is: "
+ element);
        } else {
            System.out.println("Invalid index! Please enter index
between 0 and " + (colors.size() - 1));
        }

        sc.close();
    }
}
```

#output



48. Write a Java program to update an ArrayList element by a given element.

Ans :

```
import java.util.ArrayList;
import java.util.Scanner;
public class C48_UpdateArrayListElement {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number
        ArrayList<String> colors = new ArrayList<>();
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");
        colors.add("Yellow");
        colors.add("Orange");
        Scanner sc = new Scanner(System.in);
        System.out.println("Original list: " + colors);
        System.out.print("Enter the index of element to update: ");
        int index = sc.nextInt();
        sc.nextLine();  // consume newline
        if (index >= 0 && index < colors.size()) {
            System.out.print("Enter new color to update: ");
            String newColor = sc.nextLine();

            colors.set(index, newColor);  // Update element

            System.out.println("Updated list: " + colors);
        } else {
            System.out.println("Invalid index! Please enter index
between 0 and " + (colors.size() - 1));
        }
        sc.close();
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

 javac C48_UpdateArrayListElement.java && java C48_UpdateArrayListElement
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Original list: [Red, Green, Blue, Yellow, Orange]
Enter the index of element to update: 3
Enter new color to update: Aqua
Updated list: [Red, Green, Blue, Aqua, Orange]
```

49. Write a Java program to remove the third element from an ArrayList.

Ans :

```java
import java.util.ArrayList;

public class C49_RemoveThirdElement {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>();
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");
        colors.add("Yellow");
        colors.add("Orange");

        System.out.println("Original list: " + colors);

        if (colors.size() >= 3) {
            colors.remove(2);  // Remove element at index 2 (third
element)
            System.out.println("After removing third element: " +
colors);
        } else {
            System.out.println("List has less than 3 elements,
cannot remove third element.");
        }
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code  +  ∨

                                                                   &
 javac C49_RemoveThirdElement.java && java C49_RemoveThirdElement
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Original list: [Red, Green, Blue, Yellow, Orange]
After removing third element: [Red, Green, Yellow, Orange]

c:\Users\housh\Desktop\vivek\java>
```

50. Write a Java program to search for an element in an ArrayList.

Ans :

```
import java.util.ArrayList;
import java.util.Scanner;

public class C50_SearchElementInArrayList {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>();
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");
        colors.add("Yellow");
        colors.add("Orange");

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter color to search: ");
        String searchColor = sc.nextLine();

        if (colors.contains(searchColor)) {
            System.out.println(searchColor + " is found in the
list.");
        } else {
            System.out.println(searchColor + " is NOT found in the
list.");
        }

        sc.close();
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

 javac C50_SearchElementInArrayList.java && java C50_SearchElementInArrayLis
t
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Enter color to search: Red
Red is found in the list.

c:\Users\houshk\Desktop\vivek\java>
```

51. Write a Java program to sort a given ArrayList.

Ans :

```
import java.util.ArrayList;
import java.util.Collections;

public class C51_SortArrayList {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>();
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");
        colors.add("Yellow");
        colors.add("Orange");

        System.out.println("Original list: " + colors);

        // Sorting the ArrayList
        Collections.sort(colors);

        System.out.println("Sorted list: " + colors);
    }
}
```

#output

52. Write a Java program to copy one array list into another.

Ans :

```
import java.util.ArrayList;
import java.util.Collections;

public class C52_CopyArrayList {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        // Source ArrayList
        ArrayList<String> sourceList = new ArrayList<>();
        sourceList.add("Red");
        sourceList.add("Green");
        sourceList.add("Blue");
        // Destination ArrayList (should be at least the same size
as urceList)
        ArrayList<String> destinationList = new ArrayList<>();
        destinationList.add("White");
        destinationList.add("Black");
        destinationList.add("Gray");

        System.out.println("Source List: " + sourceList);
        System.out.println("Destination List (before copy): " +
destinationList);

        // Copy sourceList into destinationList
        Collections.copy(destinationList, sourceList);

        System.out.println("Destination List (after copy): " +
destinationList);
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshu\Desktop\vivek\java>javac C52_CopyArrayList.java &&  java C52_
CopyArrayList
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Source List: [Red, Green, Blue]
Destination List (before copy): [White, Black, Gray]
Destination List (after copy): [Red, Green, Blue]

c:\Users\houshu\Desktop\vivek\java>
```

53. Write a Java program to shuffle elements in an array list.

Ans :

```
import java.util.ArrayList;
import java.util.Collections;

public class C53_ShuffleArrayList {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>();
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");
        colors.add("Yellow");
        colors.add("Orange");

        System.out.println("Original List: " + colors);

        // Shuffling the list
        Collections.shuffle(colors);

        System.out.println("Shuffled List: " + colors);
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshu\Desktop\vivek\java>javac C53_ShuffleArrayList.java && java C5
3_ShuffleArrayList
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Original List: [Red, Green, Blue, Yellow, Orange]
Shuffled List: [Orange, Yellow, Green, Red, Blue]

c:\Users\houshu\Desktop\vivek\java>
```

54. Write a Java program to reverse elements in an array list.

Ans :

```
import java.util.ArrayList;
import java.util.Collections;

public class C54_ReverseArrayList {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>();
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");
        colors.add("Yellow");
        colors.add("Orange");

        System.out.println("Original List: " + colors);

        // Reversing the list
        Collections.reverse(colors);

        System.out.println("Reversed List: " + colors);
    }
}
```

#output

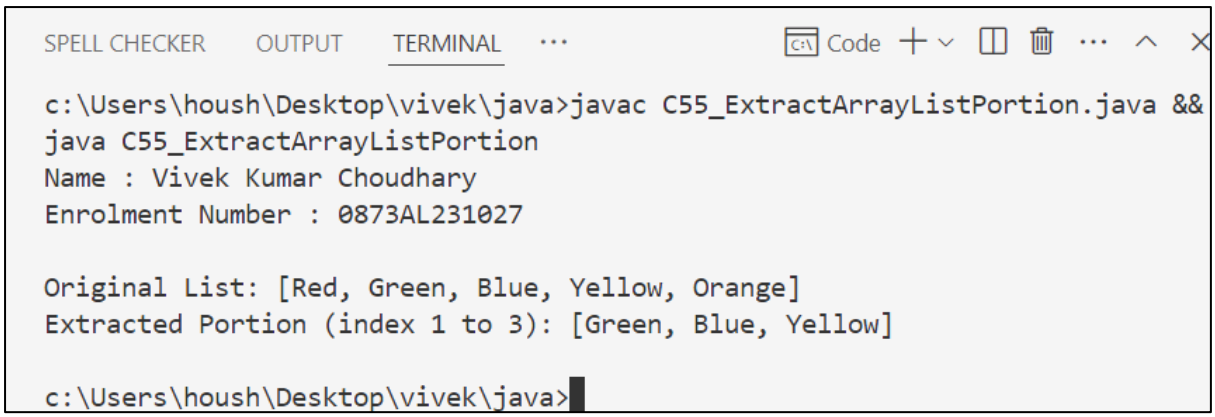
```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshu\Desktop\vivek\java>javac C54_ReverseArrayList.java && java C5
4_ReverseArrayList
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Original List: [Red, Green, Blue, Yellow, Orange]
Reversed List: [Orange, Yellow, Blue, Green, Red]

c:\Users\houshu\Desktop\vivek\java>
```

55. Write a Java program to extract a portion of an array list.

Ans :

```
import java.util.ArrayList;
import java.util.List;

public class C55_ExtractArrayListPortion {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>();
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");
        colors.add("Yellow");
        colors.add("Orange");

        System.out.println("Original List: " + colors);

        // Extracting a portion (from index 1 to 4, exclusive of 4)
        List<String> subList = colors.subList(1, 4);

        System.out.println("Extracted Portion (index 1 to 3): " +
subList);
    }
}
```

#output



56. Write a Java program to compare two array lists.

Ans :

```
import java.util.ArrayList;

public class C56_CompareArrayLists {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> list1 = new ArrayList<>();
        list1.add("Red");
        list1.add("Green");
        list1.add("Blue");

        ArrayList<String> list2 = new ArrayList<>();
        list2.add("Red");
        list2.add("Green");
        list2.add("Blue");

        ArrayList<String> list3 = new ArrayList<>();
        list3.add("Green");
        list3.add("Red");
        list3.add("Blue");

        // Compare list1 and list2
        boolean isEqual12 = list1.equals(list2);
        System.out.println("list1 equals list2: " + isEqual12);

        // Compare list1 and list3
        boolean isEqual13 = list1.equals(list3);
        System.out.println("list1 equals list3: " + isEqual13);
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshu\Desktop\vivek\java>javac C56_CompareArrayLists.java && java C
56_CompareArrayLists
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

list1 equals list2: true
list1 equals list3: false

c:\Users\houshu\Desktop\vivek\java>
```

57. Write a Java program that swaps two elements in an array list.

Ans :

```
import java.util.ArrayList;
import java.util.Collections;

public class C57_SwapArrayListElements {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>();
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");
        colors.add("Yellow");

        System.out.println("Original List: " + colors);

        // Swapping elements at index 1 and 3
        Collections.swap(colors, 1, 3);

        System.out.println("List after swapping index 1 and 3: " +
colors);
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshu\Desktop\vivek\java>javac C57_SwapArrayListElements.java && ja
va C57_SwapArrayListElements
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Original List: [Red, Green, Blue, Yellow]
List after swapping index 1 and 3: [Red, Yellow, Blue, Green]

c:\Users\houshu\Desktop\vivek\java>
```

58. Write a Java program to join two array lists.

Ans :

```
import java.util.ArrayList;

public class C58_JoinArrayLists {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        // First ArrayList
        ArrayList<String> list1 = new ArrayList<>();
        list1.add("Red");
        list1.add("Green");
        list1.add("Blue");

        // Second ArrayList
        ArrayList<String> list2 = new ArrayList<>();
        list2.add("Yellow");
        list2.add("Orange");
        list2.add("Purple");

        System.out.println("List 1: " + list1);
        System.out.println("List 2: " + list2);

        // Joining list2 into list1
        list1.addAll(list2);

        System.out.println("Joined List: " + list1);
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshl\Desktop\vivek\java>javac C58_JoinArrayLists.java && java C58_
JoinArrayLists
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

List 1: [Red, Green, Blue]
List 2: [Yellow, Orange, Purple]
Joined List: [Red, Green, Blue, Yellow, Orange, Purple]

c:\Users\houshl\Desktop\vivek\java>
```

59. Write a Java program to clone an array list to another array list.

Ans :

```
import java.util.ArrayList;

public class C59_CloneArrayList {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> originalList = new ArrayList<>();
        originalList.add("Red");
        originalList.add("Green");
        originalList.add("Blue");

        // Cloning the ArrayList
        ArrayList<String> clonedList = (ArrayList<String>)
originalList.clone();

        System.out.println("Original List: " + originalList);
        System.out.println("Cloned List: " + clonedList);
    }
}
```

#output

```
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Original List: [Red, Green, Blue]
Cloned List: [Red, Green, Blue]

C:\Users\houshl\Desktop\vivek\java>
```

60. Write a Java program to empty an array list.

Ans :

```
import java.util.ArrayList;

public class C60_EmptyArrayList {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>();
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");

        System.out.println("Before emptying: " + colors);

        // Emptying the list
        colors.clear();

        System.out.println("After emptying: " + colors);
    }
}
```

#output

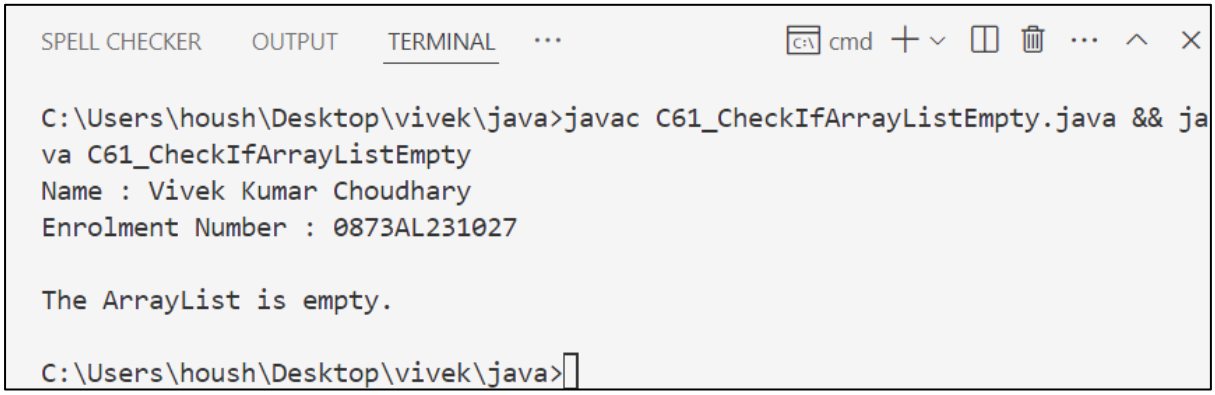
```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshl\Desktop\vivek\java>javac C60_EmptyArrayList.java && java C60_
EmptyArrayList
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Before emptying: [Red, Green, Blue]
After emptying: []

C:\Users\houshl\Desktop\vivek\java>
```

61. Write a Java program to test whether an array list is empty or not.

Ans :

```
import java.util.ArrayList;

public class C61_CheckIfArrayListEmpty {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>();

        // Check if list is empty
        if (colors.isEmpty()) {
            System.out.println("The ArrayList is empty.");
        } else {
            System.out.println("The ArrayList is not empty.");
        }
    }
}
```

#output



62. Write a Java program for trimming the capacity of an array list.

Ans :

```
import java.util.ArrayList;

public class C62_TrimArrayList {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>(20); // initial
capacity 20
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");

        System.out.println("Size before trimming: " +
colors.size());

        // Trimming capacity to the current size
        colors.trimToSize();

        System.out.println("Size after trimming: " + colors.size());
    }
}
```

#output

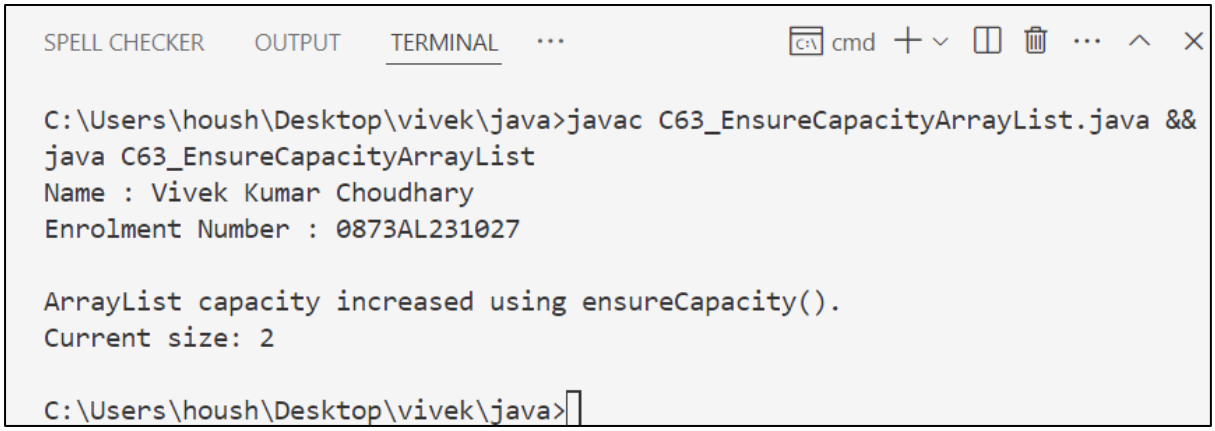
```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshub\Desktop\vivek\java>javac C62_TrimArrayList.java && java C62_T
rimArrayList
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Size before trimming: 3
Size after trimming: 3
```

63. Write a Java program to increase an array list size.

Ans :

```
import java.util.ArrayList;

public class C63_EnsureCapacityArrayList {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>();
        colors.add("Red");
        colors.add("Green");

        // Increasing the capacity to store at least 10 elements
        colors.ensureCapacity(10);

        System.out.println("ArrayList capacity increased using
ensureCapacity().");
        System.out.println("Current size: " + colors.size());
    }
}
```

#output



64. Write a Java program to replace the second element of an ArrayList with the specified element.

Ans :

```
import java.util.ArrayList;

public class C64_ReplaceSecondElement {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>();
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");

        System.out.println("Original List: " + colors);

        // Replacing second element (index 1)
        colors.set(1, "Yellow");

        System.out.println("List after replacement: " + colors);
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshu\Desktop\vivek\java>javac C64_ReplaceSecondElement.java && jav
a C64_ReplaceSecondElement
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Original List: [Red, Green, Blue]
List after replacement: [Red, Yellow, Blue]

C:\Users\houshu\Desktop\vivek\java>
```

65. Write a Java program to print all the elements of an ArrayList using the elements' position.

Ans :

```
import java.util.ArrayList;

public class C65_PrintElementsByPosi66tion {
    public static void main(String[] args) {
        Name.info();  // Print name and enrollment number

        ArrayList<String> colors = new ArrayList<>();
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");
        colors.add("Yellow");

        System.out.println("Elements in the ArrayList:");

        // Using for loop with index to print elements by position
        for (int i = 0; i < colors.size(); i++) {
            System.out.println("Element at position " + i + ": " +
colors.get(i));
        }
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshu\Desktop\vivek\java>javac C65_PrintElementsByPosition.java &&
java C65_PrintElementsByPosition
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Elements in the ArrayList:
Element at position 0: Red
Element at position 1: Green
Element at position 2: Blue
Element at position 3: Yellow

C:\Users\houshu\Desktop\vivek\java>
```

66. Write a Java program to append a specified element to the end of a linked list.

Ans :

```
import java.util.LinkedList;

public class C66_AppendToLinkedList {
    public static void main(String[] args) {
        Name.info();  // method to print name and enrollment number

        LinkedList<String> colors = new LinkedList<>();
        colors.add("Red");
        colors.add("Green");
        colors.add("Blue");

        System.out.println("Original LinkedList: " + colors);

        // Append element at end
        colors.addLast("Yellow");

        System.out.println("After appending: " + colors);
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshu\Desktop\vivek\java>javac C66_AppendToLinkedList.java && java
C66_AppendToLinkedList
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Original LinkedList: [Red, Green, Blue]
After appending: [Red, Green, Blue, Yellow]

C:\Users\houshu\Desktop\vivek\java>
```

67. Write a Java program to iterate through all elements in a linked list.

Ans :

```
import java.util.LinkedList;

public class C67_IterateLinkedList {
    public static void main(String[] args) {
        Name.info();  // method to print name and enrollment number

        LinkedList<String> names = new LinkedList<>();
        names.add("Alice");
        names.add("Bob");
        names.add("Charlie");

        System.out.println("Iterating through LinkedList:");

        for (String name : names) {
            System.out.println(name);
        }
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshl\Desktop\vivek\java>javac C67_IterateLinkedList.java && java C
67_IterateLinkedList
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Iterating through LinkedList:
Alice
Bob
Charlie

C:\Users\houshl\Desktop\vivek\java>
```

68. Write a Java program to iterate through all elements starting from a specified position in a linked list.

Ans :

```
import java.util.LinkedList;

public class C68_IterateFromPosition {
    public static void main(String[] args) {
        Name.info();  // method to print name and enrollment number

        LinkedList<String> fruits = new LinkedList<>();
        fruits.add("Apple");
        fruits.add("Banana");
        fruits.add("Cherry");
        fruits.add("Date");

        int startIndex = 2; // starting from index 2

        System.out.println("Iterating from index " + startIndex +
":");

        for (int i = startIndex; i < fruits.size(); i++) {
            System.out.println("Element at index " + i + ": " +
fruits.get(i));
        }
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshl\Desktop\vivek\java>javac C68_IterateFromPosition.java && java
 C68_IterateFromPosition
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Iterating from index 2:
Element at index 2: Cherry
Element at index 3: Date

C:\Users\houshl\Desktop\vivek\java>
```

69. Write a Java program to iterate a linked list in reverse order.

Ans :

```
import java.util.LinkedList;
import java.util.Iterator;

public class C69_ReverseIterateLinkedList {
    public static void main(String[] args) {
        Name.info();  // method to print name and enrollment number

        LinkedList<String> cities = new LinkedList<>();
        cities.add("Delhi");
        cities.add("Mumbai");
        cities.add("Chennai");
        cities.add("Kolkata");

        System.out.println("Iterating in reverse order:");

        Iterator<String> itr = cities.descendingIterator();
        while (itr.hasNext()) {
            System.out.println(itr.next());
        }
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshl\Desktop\vivek\java>javac C69_ReverseIterateLinkedList.java &&
 java  C69_ReverseIterateLinkedList
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Iterating in reverse order:
Kolkata
Chennai
Mumbai
Delhi

C:\Users\houshl\Desktop\vivek\java>
```

70. Write a Java program to insert a specified element at a given position in a linked list.

Ans :

```
import java.util.LinkedList;

public class C70_InsertAtPosition {
    public static void main(String[] args) {
        Name.info();  // method to print name and enrollment number

        LinkedList<String> animals = new LinkedList<>();
        animals.add("Dog");
        animals.add("Cat");
        animals.add("Elephant");

        System.out.println("Original LinkedList: " + animals);

        // Insert "Lion" at index 1
        animals.add(1, "Lion");

        System.out.println("After inserting at position 1: " +
animals);
    }
}
```

#output

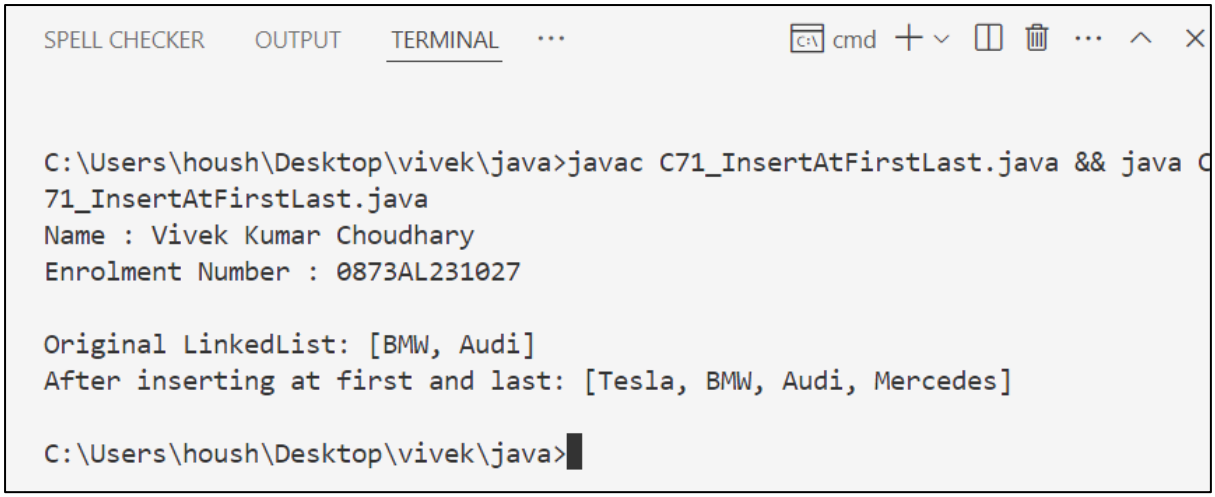
```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshl\Desktop\vivek\java>javac C70_InsertAtPosition.java && java C7
0_InsertAtPosition.java
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Original LinkedList: [Dog, Cat, Elephant]
After inserting at position 1: [Dog, Lion, Cat, Elephant]

C:\Users\houshl\Desktop\vivek\java>
```

71. Write a Java program to insert elements at the first and last positions of a linked list.

Ans :

```
import java.util.LinkedList;

public class C71_InsertAtFirstLast {
    public static void main(String[] args) {
        Name.info();  // method to print name and enrollment number

        LinkedList<String> cars = new LinkedList<>();
        cars.add("BMW");
        cars.add("Audi");

        System.out.println("Original LinkedList: " + cars);

        // Insert at first and last positions
        cars.addFirst("Tesla");
        cars.addLast("Mercedes");

        System.out.println("After inserting at first and last: " +
cars);
    }
}
```

#output



72. Write a Java program to add all elements from one TreeSet to another TreeSet.

Ans :

```
import java.util.TreeSet;

public class C72_AddTreeSets {
    public static void main(String[] args) {
        Name.info();  // method to print name and enrollment number

        TreeSet<String> set1 = new TreeSet<>();
        set1.add("Red");
        set1.add("Green");
        set1.add("Blue");

        TreeSet<String> set2 = new TreeSet<>();
        set2.addAll(set1);

        System.out.println("First TreeSet: " + set1);
        System.out.println("Second TreeSet after adding: " + set2);
    }
}
```

#output

73. Write a Java program to display the elements of a TreeSet in reverse order.

Ans :

```
import java.util.TreeSet;
import java.util.Iterator;

public class C73_ReverseTreeSet {
    public static void main(String[] args) {
        Name.info();  // method to print name and enrollment number

        TreeSet<Integer> numbers = new TreeSet<>();
        numbers.add(10);
        numbers.add(5);
        numbers.add(20);
        numbers.add(15);

        Iterator<Integer> reverseItr = numbers.descendingIterator();
        System.out.println("TreeSet in reverse order:");
        while (reverseItr.hasNext()) {
            System.out.println(reverseItr.next());
        }
    }
}
```

#output

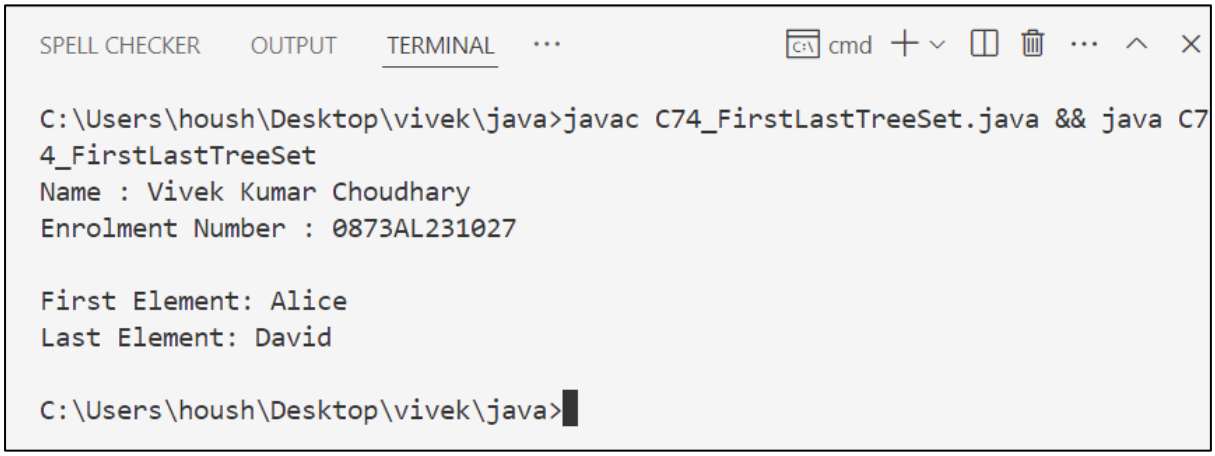
```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshu\Desktop\vivek\java>javac C73_ReverseTreeSet.java && java C73_
ReverseTreeSet
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

TreeSet in reverse order:
20
15
10
5
```

74. Write a Java program to retrieve the first and last elements from a TreeSet.

Ans :

```
import java.util.TreeSet;

public class C74_FirstLastTreeSet {
    public static void main(String[] args) {
        Name.info();  // method to print name and enrollment number

        TreeSet<String> names = new TreeSet<>();
        names.add("Alice");
        names.add("Bob");
        names.add("Charlie");
        names.add("David");

        System.out.println("First Element: " + names.first());
        System.out.println("Last Element: " + names.last());
    }
}
```

#output



75. Write a Java program to clone a TreeSet into another TreeSet

Ans :

```
import java.util.TreeSet;

public class C75_CloneTreeSet {
    public static void main(String[] args) {
        Name.info();  // method to print name and enrollment number

        TreeSet<String> original = new TreeSet<>();
        original.add("One");
        original.add("Two");
        original.add("Three");

        TreeSet<String> cloned = (TreeSet<String>) original.clone();

        System.out.println("Original TreeSet: " + original);
        System.out.println("Cloned TreeSet: " + cloned);
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshub\Desktop\vivek\java>javac C75_CloneTreeSet.java && java C75_Cl
oneTreeSet
Note: C75_CloneTreeSet.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Original TreeSet: [One, Three, Two]
Cloned TreeSet: [One, Three, Two]

C:\Users\houshub\Desktop\vivek\java>
```

76. Write a Java program to count the number of elements in a TreeSet.

Ans :

```
import java.util.TreeSet;

public class C76_CountTreeSet {
    public static void main(String[] args) {
        Name.info();  // method to print name and enrollment number

        TreeSet<Integer> numbers = new TreeSet<>();
        numbers.add(10);
        numbers.add(20);
        numbers.add(30);
        numbers.add(40);

        System.out.println("Total number of elements: " +
numbers.size());
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshl\Desktop\vivek\java>javac C76_CountTreeSet.java && java C76_Co
untTreeSet
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Total number of elements: 4

C:\Users\houshl\Desktop\vivek\java>
```

77. Write a Java program to compare two TreeSets.

Ans :

```
import java.util.TreeSet;

public class C77_CompareTreeSets {
    public static void main(String[] args) {
        Name.info();  // method to print name and enrollment number

        TreeSet<String> set1 = new TreeSet<>();
        set1.add("A");
        set1.add("B");
        set1.add("C");

        TreeSet<String> set2 = new TreeSet<>();
        set2.add("B");
        set2.add("C");
        set2.add("D");

        for (String element : set1) {
            if (set2.contains(element)) {
                System.out.println(element + " is present in both
sets.");
            } else {
                System.out.println(element + " is not present in
both sets.");
            }
        }
    }
}
```

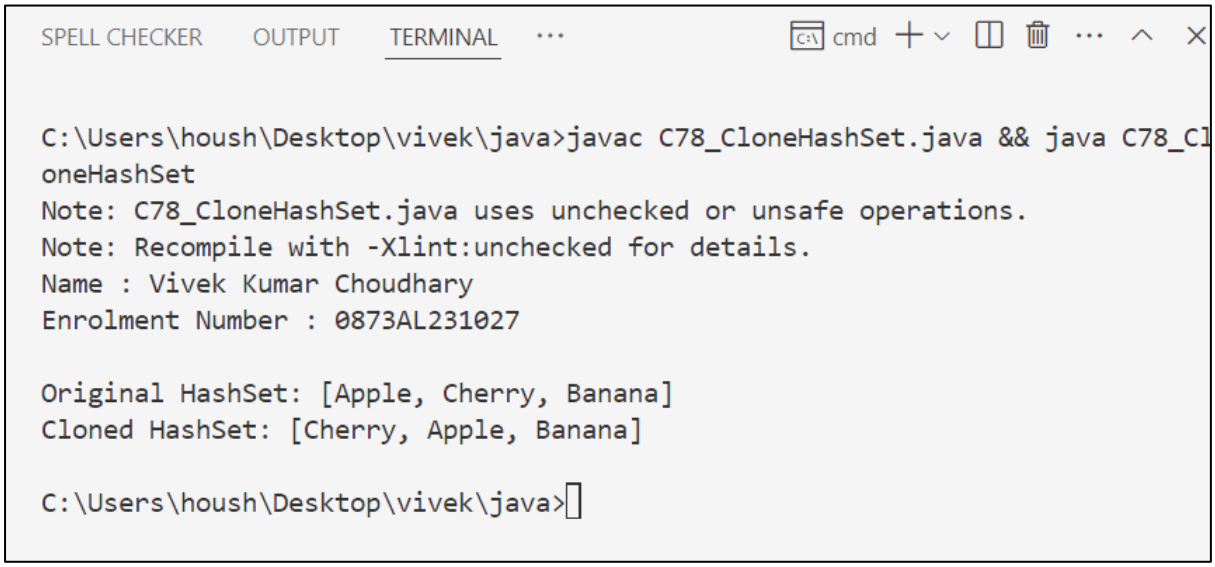
#output

78. Write a Java program to clone one HashSet into another.

Ans :

```
import java.util.HashSet;

public class C78_CloneHashSet {
    public static void main(String[] args) {
        Name.info();  // method to print name and enrollment number

        HashSet<String> original = new HashSet<>();
        original.add("Apple");
        original.add("Banana");
        original.add("Cherry");

        HashSet<String> clone = (HashSet<String>) original.clone();

        System.out.println("Original HashSet: " + original);
        System.out.println("Cloned HashSet: " + clone);
    }
}
```

#output



79. Write a Java program to convert a HashSet into an array.

Ans :

```
import java.util.HashSet;

public class C79_HashSetToArray {
    public static void main(String[] args) {
        Name.info();

        HashSet<String> set = new HashSet<>();
        set.add("Red");
        set.add("Green");
        set.add("Blue");

        String[] array = new String[set.size()];
        set.toArray(array);

        System.out.println("Array elements:");
        for (String color : array) {
            System.out.println(color);
        }
    }
}
```

#output

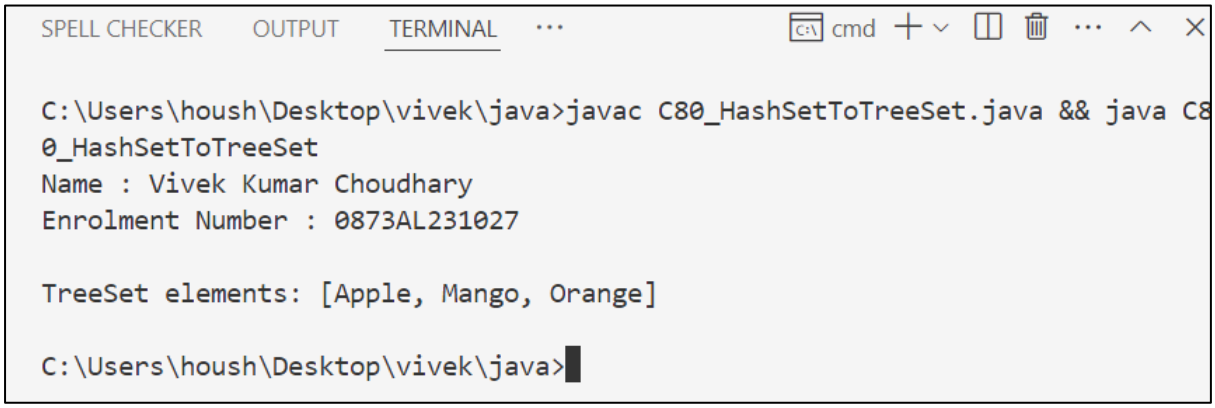
```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshu\Desktop\vivek\java>javac C79_HashSetToArray.java && java C79_
HashSetToArray
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Array elements:
Red
Blue
Green

C:\Users\houshu\Desktop\vivek\java>
```

80. Write a Java program to convert a HashSet into a TreeSet.

Ans :

```
import java.util.HashSet;
import java.util.TreeSet;

public class C80_HashSetToTreeSet {
    public static void main(String[] args) {
        Name.info();

        HashSet<String> hashSet = new HashSet<>();
        hashSet.add("Orange");
        hashSet.add("Mango");
        hashSet.add("Apple");

        TreeSet<String> treeSet = new TreeSet<>(hashSet);
        System.out.println("TreeSet elements: " + treeSet);
    }
}
```

#output



81. Write a Java program to find numbers less than 7 in a TreeSet.

Ans :

```
import java.util.TreeSet;

public class C81_LessThanSeven {
    public static void main(String[] args) {
        Name.info();

        TreeSet<Integer> numbers = new TreeSet<>();
        numbers.add(2);
        numbers.add(5);
        numbers.add(8);
        numbers.add(1);
        numbers.add(10);

        System.out.println("Numbers less than 7:");
        for (Integer num : numbers.headSet(7)) {
            System.out.println(num);
        }
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshl\Desktop\vivek\java>javac C81_LessThanSeven.java && java C81_L
essThanSeven
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Numbers less than 7:
1
2
5

C:\Users\houshl\Desktop\vivek\java>
```

82. Write a Java program to compare two HashSets.

Ans :

```
import java.util.HashSet;

public class C82_CompareHashSets {
    public static void main(String[] args) {
        Name.info();

        HashSet<String> set1 = new HashSet<>();
        set1.add("Java");
        set1.add("Python");
        set1.add("C++");

        HashSet<String> set2 = new HashSet<>();
        set2.add("Python");
        set2.add("C++");
        set2.add("Java");

        if (set1.equals(set2)) {
            System.out.println("HashSets are equal.");
        } else {
            System.out.println("HashSets are not equal.");
        }
    }
}
```

#output

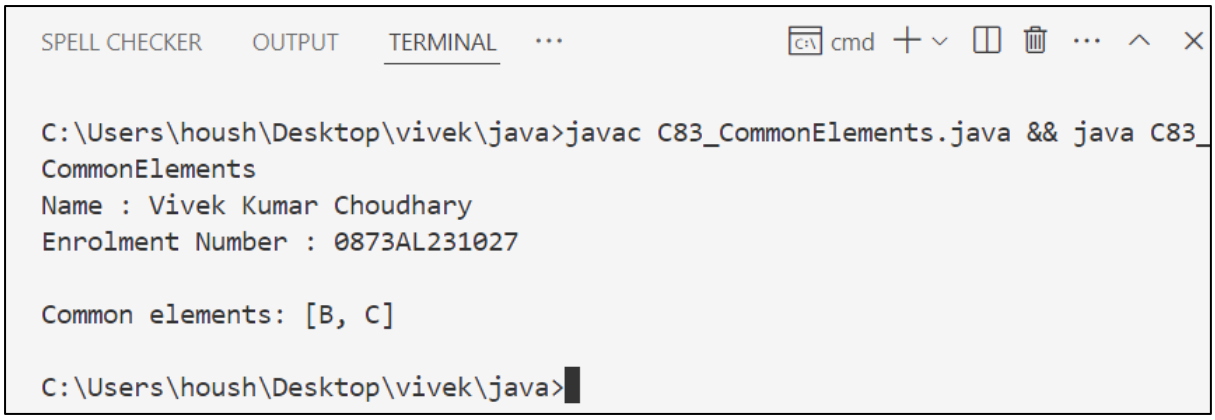
```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshu\Desktop\vivek\java>javac C82_CompareHashSets.java && java C82
_CompareHashSets
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

HashSets are equal.

C:\Users\houshu\Desktop\vivek\java>
```

83. Write a Java program to retain common elements from two sets.

Ans :

```
import java.util.HashSet;

public class C83_CommonElements {
    public static void main(String[] args) {
        Name.info();

        HashSet<String> set1 = new HashSet<>();
        set1.add("A");
        set1.add("B");
        set1.add("C");

        HashSet<String> set2 = new HashSet<>();
        set2.add("B");
        set2.add("C");
        set2.add("D");

        set1.retainAll(set2);
        System.out.println("Common elements: " + set1);
    }
}
```

#output

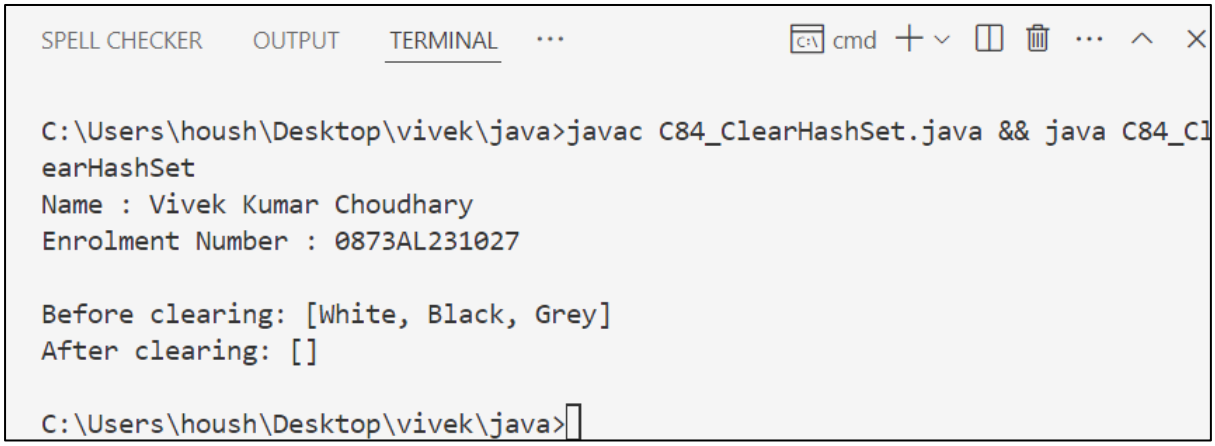


84. Write a Java program to remove all elements from a HashSet.

Ans :

```
import java.util.HashSet;

public class C84_ClearHashSet {
    public static void main(String[] args) {
        Name.info();

        HashSet<String> colors = new HashSet<>();
        colors.add("White");
        colors.add("Black");
        colors.add("Grey");

        System.out.println("Before clearing: " + colors);
        colors.clear();
        System.out.println("After clearing: " + colors);
    }
}
```

#output



85. Write a Java program to copy all mappings from one map to another.

Ans :

```
import java.util.HashMap;
import java.util.Map;

public class C85_CopyMap {
    public static void main(String[] args) {
        Name.info();  // Method to print name and enrollment number

        Map<Integer, String> map1 = new HashMap<>();
        map1.put(1, "Apple");
        map1.put(2, "Banana");
        map1.put(3, "Cherry");

        Map<Integer, String> map2 = new HashMap<>();
        map2.putAll(map1);  // Copy all mappings

        System.out.println("Original Map: " + map1);
        System.out.println("Copied Map: " + map2);
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshl\Desktop\vivek\java>javac C85_CopyMap.java && java C85_CopyMap

Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Original Map: {1=Apple, 2=Banana, 3=Cherry}
Copied Map: {1=Apple, 2=Banana, 3=Cherry}

C:\Users\houshl\Desktop\vivek\java>
```

86. Write a Java program to remove all key-value pairs from a map.

Ans :

```
import java.util.HashMap;
import java.util.Map;

public class C86_ClearMap {
    public static void main(String[] args) {
        Name.info();

        Map<String, Integer> map = new HashMap<>();
        map.put("A", 100);
        map.put("B", 200);
        map.put("C", 300);

        System.out.println("Before clearing: " + map);
        map.clear();
        System.out.println("After clearing: " + map);
    }
}
```

#output

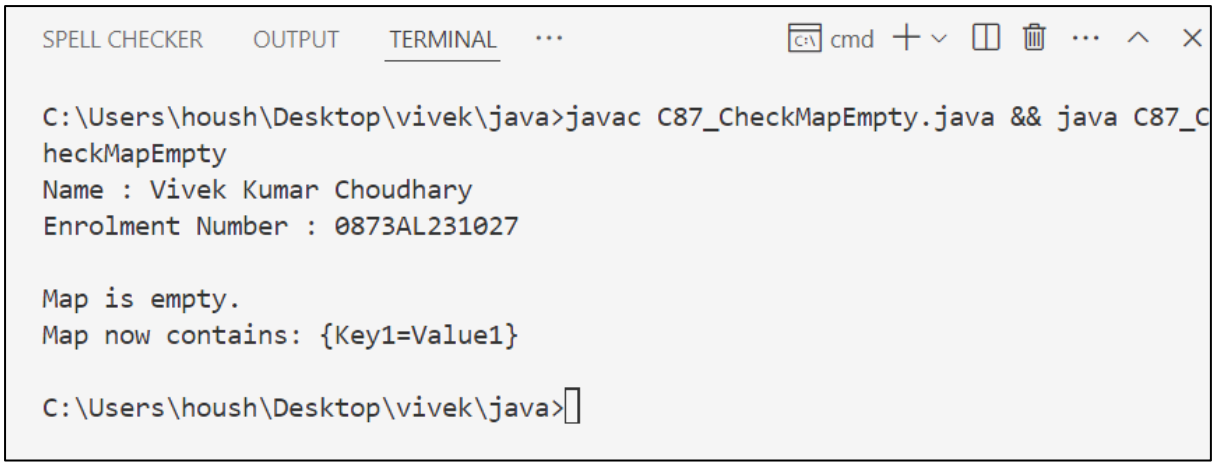
```
SPELL CHECKER   OUTPUT   TERMINAL   ···                cmd

C:\Users\houshl\Desktop\vivek\java>javac C86_ClearMap.java && java C86_ClearM
ap
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Before clearing: {A=100, B=200, C=300}
After clearing: {}

C:\Users\houshl\Desktop\vivek\java>
```

87. Write a Java program to check if a map is empty or contains key-value mappings.

Ans :

```
import java.util.HashMap;
import java.util.Map;

public class C87_CheckMapEmpty {
    public static void main(String[] args) {
        Name.info();

        Map<String, String> map = new HashMap<>();

        if (map.isEmpty()) {
            System.out.println("Map is empty.");
        } else {
            System.out.println("Map contains: " + map);
        }

        map.put("Key1", "Value1");

        if (!map.isEmpty()) {
            System.out.println("Map now contains: " + map);
        }
    }
}
```

#output

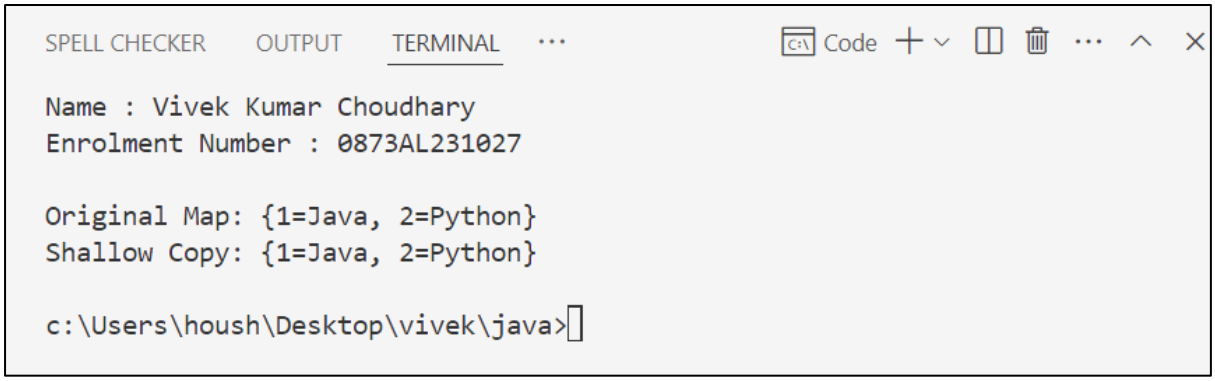


88. Write a Java program to create a shallow copy of a HashMap instance.

Ans :

```
import java.util.HashMap;

public class C88_ShallowCopyMap {
    public static void main(String[] args) {
        Name.info();

        HashMap<Integer, String> original = new HashMap<>();
        original.put(1, "Java");
        original.put(2, "Python");

        HashMap<Integer, String> shallowCopy = (HashMap<Integer,
String>) original.clone();

        System.out.println("Original Map: " + original);
        System.out.println("Shallow Copy: " + shallowCopy);
    }
}
```

#output



89. Write a Java program to test whether a specified key exists in the map.

Ans :

```
import java.util.HashMap;

public class C89_KeyExistsInMap {
    public static void main(String[] args) {
        Name.info();

        HashMap<String, Integer> map = new HashMap<>();
        map.put("Math", 90);
        map.put("Science", 85);
        map.put("English", 95);

        String keyToCheck = "Science";

        if (map.containsKey(keyToCheck)) {
            System.out.println("Key \"" + keyToCheck + "\" exists
with value: " + map.get(keyToCheck));
        } else {
            System.out.println("Key \"" + keyToCheck + "\" does not
exist.");
        }
    }
}
```

#output

```
SPELL CHECKER   OUTPUT   TERMINAL   ···                Code

c:\Users\houshub\Desktop\vivek\java>cd "c:\Users\houshub\Desktop\vivek\java\" &&
 javac C89_KeyExistsInMap.java && java C89_KeyExistsInMap
Name : Vivek Kumar Choudhary
Enrolment Number : 0873AL231027

Key "Science" exists with value: 85

c:\Users\houshub\Desktop\vivek\java>
```

90. Create a table Item_dtls (Electronics)
Try to insert at least 10 records in the above table
Try to insert at least 2 records with null value

Ans :

```
CREATE TABLE Item_dtls (
    Item_ID INT PRIMARY KEY,
    Item_Name VARCHAR(100),
    Manufacturer_ID INT,
    Price DECIMAL(10,2),
    Warranty_Years INT
);

-- Insert 10 records (2 with NULL values)
INSERT INTO Item_dtls VALUES (1, 'Smartphone', 1, 24999.99, 2);
INSERT INTO Item_dtls VALUES (2, 'Laptop', 2, 55999.00, 3);
INSERT INTO Item_dtls VALUES (3, 'Tablet', 3, 14999.50, 2);
INSERT INTO Item_dtls VALUES (4, 'Smartwatch', 4, 18999.99, 1);
INSERT INTO Item_dtls VALUES (5, 'Monitor', 5, 12999.00, 3);
INSERT INTO Item_dtls VALUES (6, 'Keyboard', 6, 1999.99, 1);
INSERT INTO Item_dtls VALUES (7, 'Mouse', 6, 899.50, 1);
INSERT INTO Item_dtls VALUES (8, 'Speakers', 7, 2499.75, 1);
INSERT INTO Item_dtls VALUES (9, NULL, 8, NULL, NULL); -- NULL
values
INSERT INTO Item_dtls VALUES (10, 'Printer', NULL, 4999.99, 2); --
NULL manufacturer
```

#output

Item_ID	Item_Name	Manufacturer_ID	Price	Warranty_Years
1	Smartphone	1	24999.99	2
2	Laptop	2	55999.00	3
3	Tablet	3	14999.50	2
4	Smartwatch	4	18999.99	1
5	Monitor	5	12999.00	3
6	Keyboard	6	1999.99	1
7	Mouse	6	899.50	1
8	Speakers	7	2499.75	1
9	*NULL*	8	*NULL*	*NULL*
10	Printer	*NULL*	4999.99	2

91. Create a table Sales_dtls
Try to insert at least 10 records in the above table
Try to insert at least 2 records with null value

Ans :

```
CREATE TABLE Sales_dtls (
    Sale_ID INT PRIMARY KEY,
    Item_ID INT,
    Customer_Name VARCHAR(100),
    Quantity INT,
    Sale_Date DATE,
    Total_Amount DECIMAL(10,2));
INSERT INTO Sales_dtls VALUES (1, 1, 'Ravi', 2, DATE '2025-06-01',
49999.98);
INSERT INTO Sales_dtls VALUES (2, 2, 'Anjali', 1, DATE '2025-06-02',
55999.00);
INSERT INTO Sales_dtls VALUES (3, 3, 'Nikhil', 3, DATE '2025-06-02',
44998.50);
INSERT INTO Sales_dtls VALUES (4, 4, 'Divya', 1, DATE '2025-06-03',
18999.99);
INSERT INTO Sales_dtls VALUES (5, 5, 'Suresh', 2, DATE '2025-06-03',
25998.00);
INSERT INTO Sales_dtls VALUES (6, 6, 'Kavita', 1, DATE '2025-06-04',
2999.75);
INSERT INTO Sales_dtls VALUES (7, 7, 'Rahul', 2, DATE '2025-06-04',
1799.00);
INSERT INTO Sales_dtls VALUES (8, 8, 'Sneha', 1, DATE '2025-06-05',
2499.00);
INSERT INTO Sales_dtls VALUES (9, NULL, 'Amit', 1, DATE '2025-06-
06', NULL); -- NULL Item_ID, Total_Amount
INSERT INTO Sales_dtls VALUES (10, 10, NULL, NULL, NULL, NULL);
```

#output

Sale_ID	Item_ID	Customer_Name	Quantity	Sale_Date	Total_Amount
1	1	Ravi	2	2025-06-01	49999.98
2	2	Anjali	1	2025-06-02	55999.00
3	3	Nikhil	3	2025-06-02	44998.50
4	4	Divya	1	2025-06-03	18999.99
5	5	Suresh	2	2025-06-03	25998.00
6	6	Kavita	1	2025-06-04	2999.75
7	7	Rahul	2	2025-06-04	1799.00
8	8	Sneha	1	2025-06-05	2499.00
9	*NULL*	Amit	1	2025-06-06	*NULL*
10	10	*NULL*	*NULL*	*NULL*	*NULL*

92. create a table manufacturers
Try to insert at least 10 records in the above table
Try to insert at least 2 records with null value

Ans :

```
CREATE TABLE manufacturers (
    Manufacturer_ID INT PRIMARY KEY,
    Manufacturer_Name VARCHAR(100),
    Country VARCHAR(50),
    Established_Year INT,
    Email VARCHAR(100));
INSERT INTO manufacturers VALUES (1, 'Samsung', 'South Korea', 1969,
'info@samsung.com');
INSERT INTO manufacturers VALUES (2, 'Sony', 'Japan', 1946,
'info@sony.com');
INSERT INTO manufacturers VALUES (3, 'LG', 'South Korea', 1958,
'support@lg.com');
INSERT INTO manufacturers VALUES (4, 'HP', 'USA', 1939,
'support@hp.com');
INSERT INTO manufacturers VALUES (5, 'Apple', 'USA', 1976,
'apple@support.com');
INSERT INTO manufacturers VALUES (6, 'Dell', 'USA', 1984,
'dell@support.com');
INSERT INTO manufacturers VALUES (7, 'Lenovo', 'China', 1984,
'contact@lenovo.com');
INSERT INTO manufacturers VALUES (8, 'Xiaomi', 'China', 2010,
'xiaomi@support.com');
INSERT INTO manufacturers VALUES (9, 'Philips', NULL, 1891, NULL); -
- NULL country and email
INSERT INTO manufacturers VALUES (10, NULL, 'Germany', NULL, NULL);
-- NULL name, year, email
```

#output

Manufacturer_ID	Manufacturer_Name	Country	Established_Year	Email
1	Samsung	South Korea	1969	info@samsung.com
2	Sony	Japan	1946	info@sony.com
3	LG	South Korea	1958	support@lg.com
4	HP	USA	1939	support@hp.com
5	Apple	USA	1976	apple@support.com
6	Dell	USA	1984	dell@support.com
7	Lenovo	China	1984	contact@lenovo.com
8	Xiaomi	China	2010	xiaomi@support.com
9	Philips	*NULL*	1891	*NULL*
10	*NULL*	Germany	*NULL*	*NULL*

93. Consider the below tables with estimated columns and then practise below questions.

CUST DTLS
CUST Act DTLS
ACT_TYPES_INFO
PROD_DTLS
EMP
DEPT

Fetch all clerks information

Ans :

```
SELECT * FROM EMP WHERE Job_Title = 'Clerk';
```

#output

Emp_ID	EName	Job_Title	Dept_ID	Salary	Hire_Date
302	Neha Mehra	Clerk	2	25000.00	2021-03-18
305	Rohan Das	Clerk	2	26000.00	2020-09-10
308	Anjali Rao	Clerk	5	25500.00	2022-04-01
309	Nikhil Arora	Clerk	6	27000.00	2023-07-07

94. Display all departments information located at CHICAGO?

Ans :

```
SELECT * FROM DEPT WHERE Location = 'Chicago';
```

#output

Dept_ID	Dept_Name	Location
2	Accounts	Chicago
5	Admin	Chicago
8	Operations	Chicago
10	*NULL*	Chicago

95. Display product details manufactured in the current year only?

Ans :

```
SELECT * FROM PROD_DTLS
WHERE YEAR(Manufacture_Date) = YEAR(CURDATE());
```

#output

Prod_ID	Prod_Name	Manufacturer_ID	Price	Manufacture_Date	Exp
201	Laptop	1	59999.99	2025-02-10	2027-02-10
202	Smartphone	2	29999.50	2025-01-15	2026-01-15
205	Tablet	1	18999.75	2025-06-01	2027-06-01
207	Mouse	6	499.50	2025-03-03	2026-03-03
208	Smartwatch	2	14999.00	2025-06-01	2026-06-01
210	Webcam	*NULL*	*NULL*	2025-06-01	*NULL*

96. Get the details of customer accounts who opened the accounts before this year?

Ans :

```
SELECT * FROM CUST_ACT_DTLS
WHERE Open_Date IS NOT NULL
  AND EXTRACT(YEAR FROM Open_Date) < YEAR(CURDATE());
```

#output

Act_ID	Cust_ID	Account_Type	Open_Date	Balance
101	1	SAVINGS	2022-01-15	15000.50
102	2	CURRENT	2023-03-20	32000.75
103	3	SALARY	2021-07-10	45000.00
104	4	TRADING	2024-01-01	18000.25
105	5	SAVINGS	2020-12-11	12000.90
106	6	SALARY	2019-06-05	38000.00
107	7	CURRENT	2024-11-12	5000.60
108	8	TRADING	2021-05-09	25000.00
109	9	*NULL*	2023-08-17	9999.99

97. Get all SALARY account details?

Ans :

```
SELECT * FROM CUST_ACT_DTLS
WHERE Account_Type = 'SALARY';
```

#output

Act_ID	Cust_ID	Account_Type	Open_Date	Balance
103	3	SALARY	2021-07-10	45000.00
106	6	SALARY	2019-06-05	38000.00

98. Display customer names and mobile numbers from the city 'Texas'?

select cname, mobile from cust_dtls where city='Texas';

Ans :

```
SELECT CName, Mobile FROM CUST_DTLS
WHERE City = 'Texas';
```

#output

CName	Mobile
Ravi Kumar	9876543210
Suresh Mehta	9234567890
Amit Bansal	9678901234

99. Get the information of Trading account?

Ans :

```
SELECT * FROM CUST_ACT_DTLS
WHERE Account_Type = 'TRADING';
```

#output

Act_ID	Cust_ID	Account_Type	Open_Date	Balance
104	4	TRADING	2024-01-01	18000.25
108	8	TRADING	2021-05-09	25000.00

100. Display only Expired product details?
select * from prod_dtls where exp<sysdate;

Ans :

```
SELECT * FROM PROD_DTLS
WHERE Exp < CURDATE();
```

#output

Prod_ID	Prod_Name	Manufacturer_ID	Price	Manufacture_Date	Exp
204	Router	4	1999.99	2023-05-05	2024-05-05
209	Speaker	7	1299.00	2022-10-10	2023-10-10