Assignment -1

Q-1:- Define Object Oriented Programming. Discribe four Pillars of OOP.

Definition of OOP:

Object Oriented Programming (OOP) is a programe parcadim that organizes softwere design around objects and class

Four Pillars of OOP.

A. Encapsulation

- Encapsulation means binding data methods together into a single unit and restricting direct access to data.

Example:-

```
class student
{ private string name; // hidden data
  public void setName (string n)
  { name = n;
  }
  public string getName()
  { return name;
  }
}
```

B. Abstraction :-

- Abstraction means hiding implementation details and showing only necessary features.

Example :-

```
abstract class Animal
{
    abstract void sound(); // only concept
}
class Dog extends Animal
{
    void sound()
    {
        System.out.println("Dog barks");
    }
}
public class main
{
    public static void main(String[] args)
    {
        Animal a = new Dog(); // abstraction
        a.sound();
    }
}
```

c. Inheritance.

- Inheritance means one class can acquire the properties and methods of another class This promotes code reuse.

Example :-

```
class vehicle
{
    void start()
    {
        System.out.println("vehicle streats");
    }
}
class car extends vehicle
{
    void drive()
    {
        System.out.println("car is driving");
    }
}
public class main
{
    public static void main(String[]args)
    {
        car c = new car();
        c.start(); // inheritance
        c.drive(); // own method.
    }
}
```

D. Polymorphism.

- Polymorphism means one name, many forms.

Example :-

1. compile polymophism

```
public class main
} public static void main(String[] args)
  {
    MathOperation m = new mathoperation();
    System.out.println(m.add(5,10));
    System.out.println(m.add(2.5, 3.5));
  }
```

2. Running Polymorphism.

```
public class main
} public static void main(string[] args)
  {
    Animal a = new cat();
    a.sound(); // cats version executed
  }
```

conclusion:-

The four pillar of OOP makes programming modulare, reusable, secure, easier to maintain. OOP is widely used in real world applications like banking systems, ecommerce plateform.

Q-2:- Describe compilation and education of a Java program with a diagram.

Introduction :- Java is a high level, object oriented and platform independent programming language. The execution of a java program involves two major steps : compilation and education.

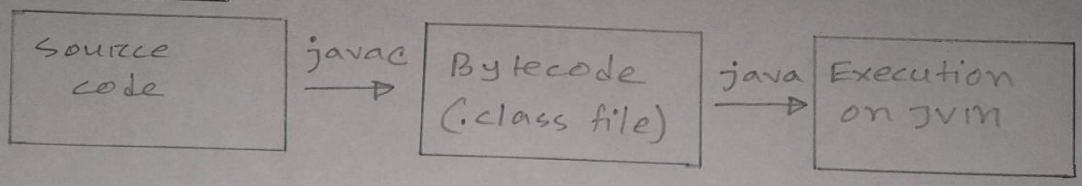
steps of compilation and Eduction :-

1. writing the program
   - Java program are written using a text editor.
   - The program is seved with the .java extension.

   Example : Hellow.java.

```
class Hellow
{
    public static void main(string[] args)
    {
        system.out.println("Hellow, Java!");
    }
}
```

Diagram :- compilation and Execution Process.



1. javac → compiler translates .java file into .class
2. java → JVM interprets the bytecode and execution
3. Java is plateform-independent because bytecode can run on any system with a JVM.

Conclution :- The process of java program execution is divided into two phases, compilation and execution. compilation converts human readable source code into platform-independent bytecode while execution is carried out by the JVM.

Q-3:- Trace the output including simulation of following problem.

* Trace and simulation of java program.

Program.

```
public class Assignment
{ public static void main(String[] args)
  {
    int x=5, z=3;
    int y= x++ - --z * --z;
    for (int i=1; i<=3; i++)
    {
      x = (x++ > y) ? (x-y) : (++y - i);
      System.out.println(x+" "+y+" | ");
    }
    System.out.println("\n Final values:
                        x = "+x+"; y="+y);
  }
}
```

2. compilation :-

javac Hellow.java

- The java compiler (javac) translates the source code into bytecode.
- The bytecode is stored in a file with a .class extension.

3. Execution :-

java Hellow

- The java virtual machine (JVM) execution the bytecode.
- JVM converts bytecodes into machine code using an interpreter or JIT (Just-In-Time) compiler.
- output.

Hellow, Java!

step-1:- Initialization

$x = 5,\ z = 3$

step-2:- Evaluate y

$y = x++ - - - z * - - z;$

1. $x++ = 5$ (then $x = 6$)
2. $- - z = 2$
3. second $- - z = 1$
4. so, $\rightarrow y = 5 - (2 * 1) = 3$

Now,

$x = 6,\ y = 3,\ z = 1$

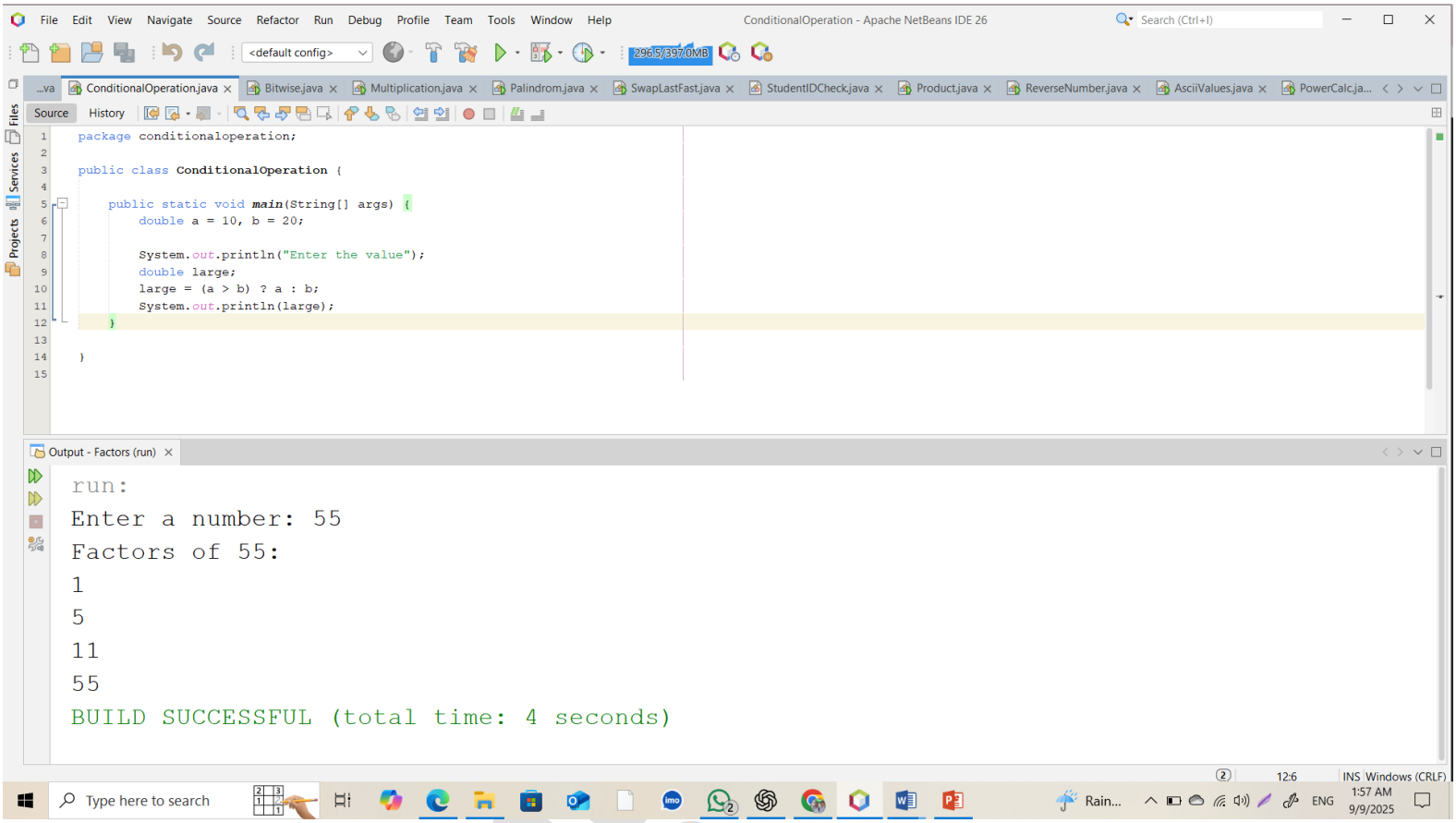
step-3:- Loop simulation

Iteration (i)	Before (x, y)	x++ > y	Action	After	Print
i = 1	(6, 3)	6 > 3 → true	x = x - y = 7 - 3 = 4	(4, 3)	4 3
i = 2	(4, 3)	4 > 3 → true	x = x - y = 5 - 3 = 2	(2, 3)	2 3
i = 3	(2, 3)	2 > 3 → false	++y - i = 4 - 3 = 1	(1, 4)	1 4

Final output

4 3 1 2 3 1 4

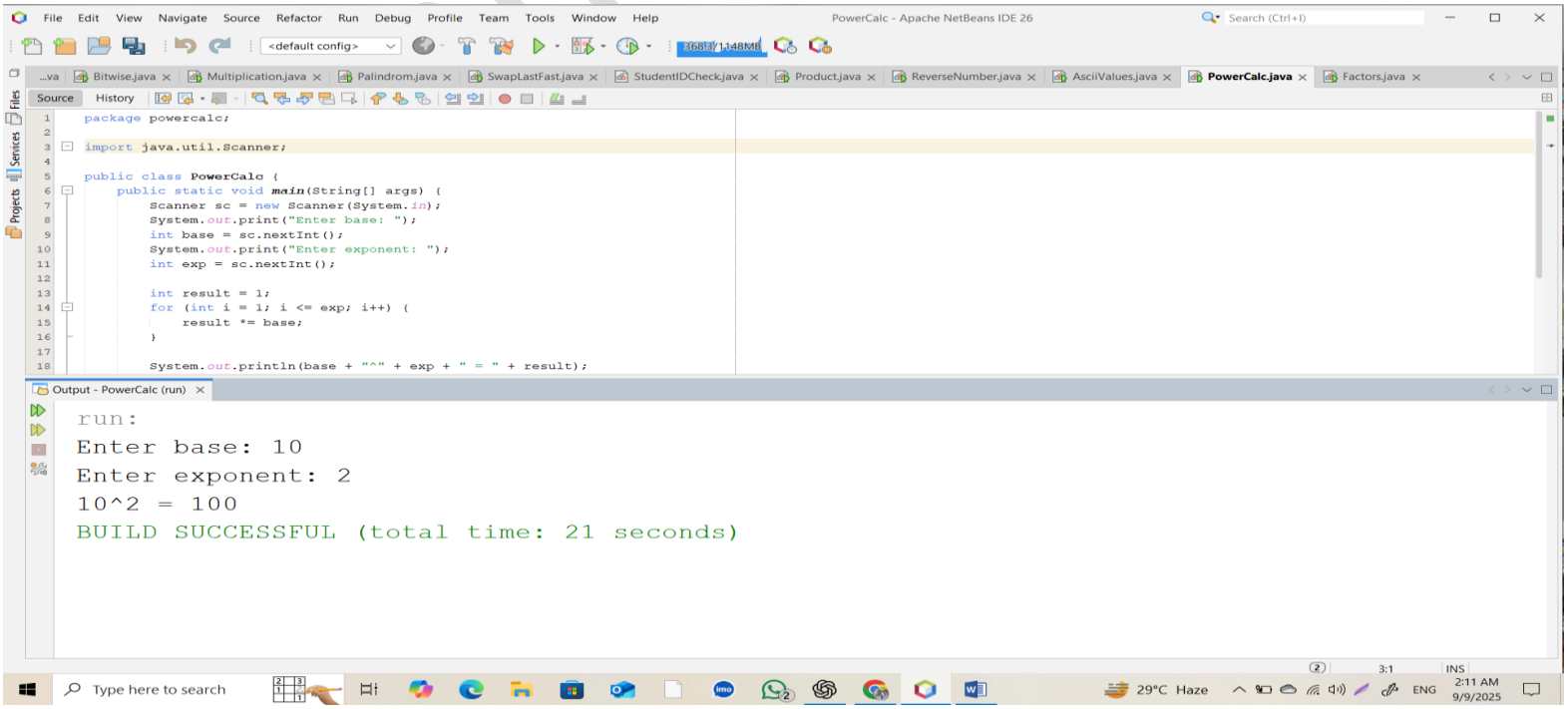
1. Print the Factor:



```
package conditionaloperation;

public class ConditionalOperation {

    public static void main(String[] args) {
        double a = 10, b = 20;

        System.out.println("Enter the value");
        double large;
        large = (a > b) ? a : b;
        System.out.println(large);
    }

}
```

```
run:
Enter a number: 55
Factors of 55:
1
5
11
55
BUILD SUCCESSFUL (total time: 4 seconds)
```

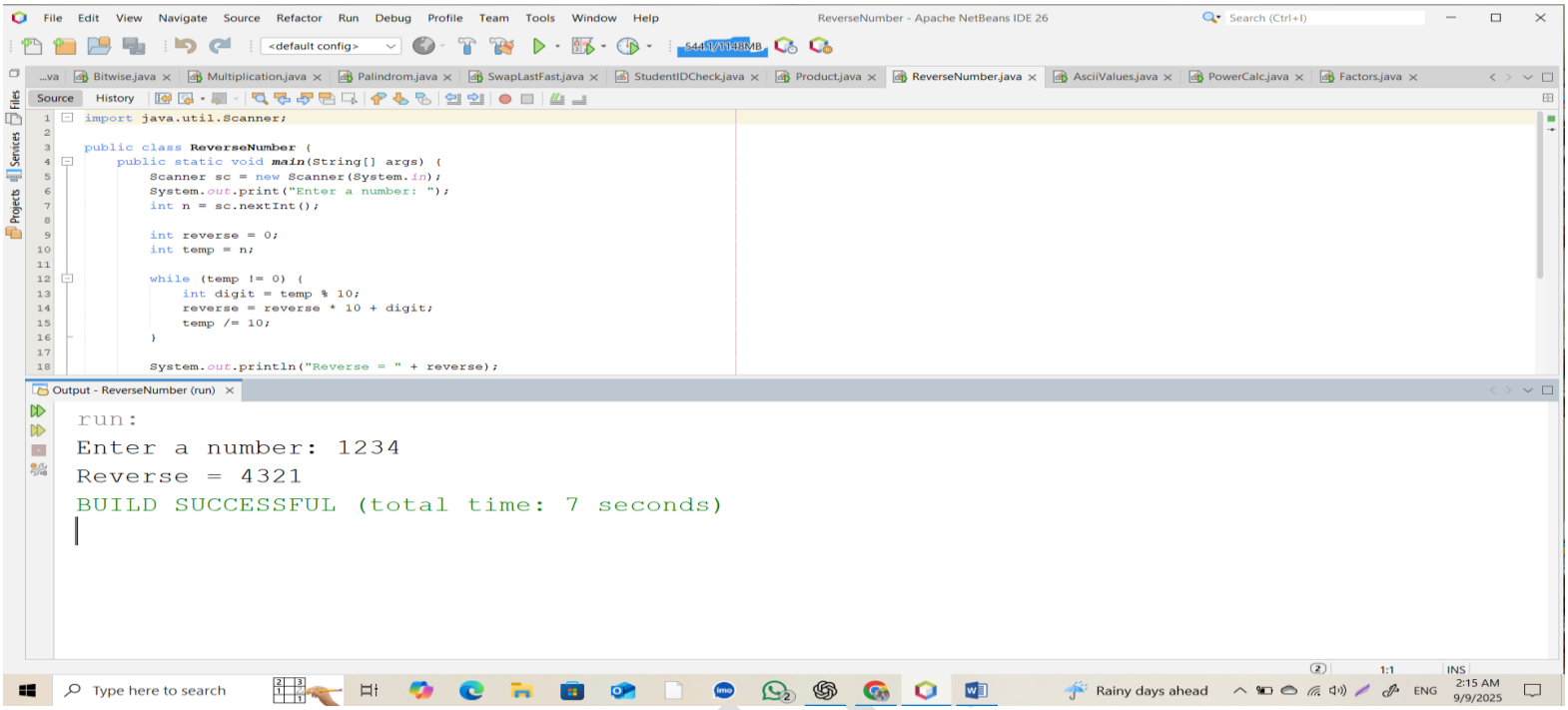
2.Print the PowerCalculate:



```
package powercalc;

import java.util.Scanner;

public class PowerCalc {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter base: ");
        int base = sc.nextInt();
        System.out.print("Enter exponent: ");
        int exp = sc.nextInt();

        int result = 1;
        for (int i = 1; i <= exp; i++) {
            result *= base;
        }

        System.out.println(base + "^" + exp + " = " + result);
    }
}
```

```
run:
Enter base: 10
Enter exponent: 2
10^2 = 100
BUILD SUCCESSFUL (total time: 21 seconds)
```

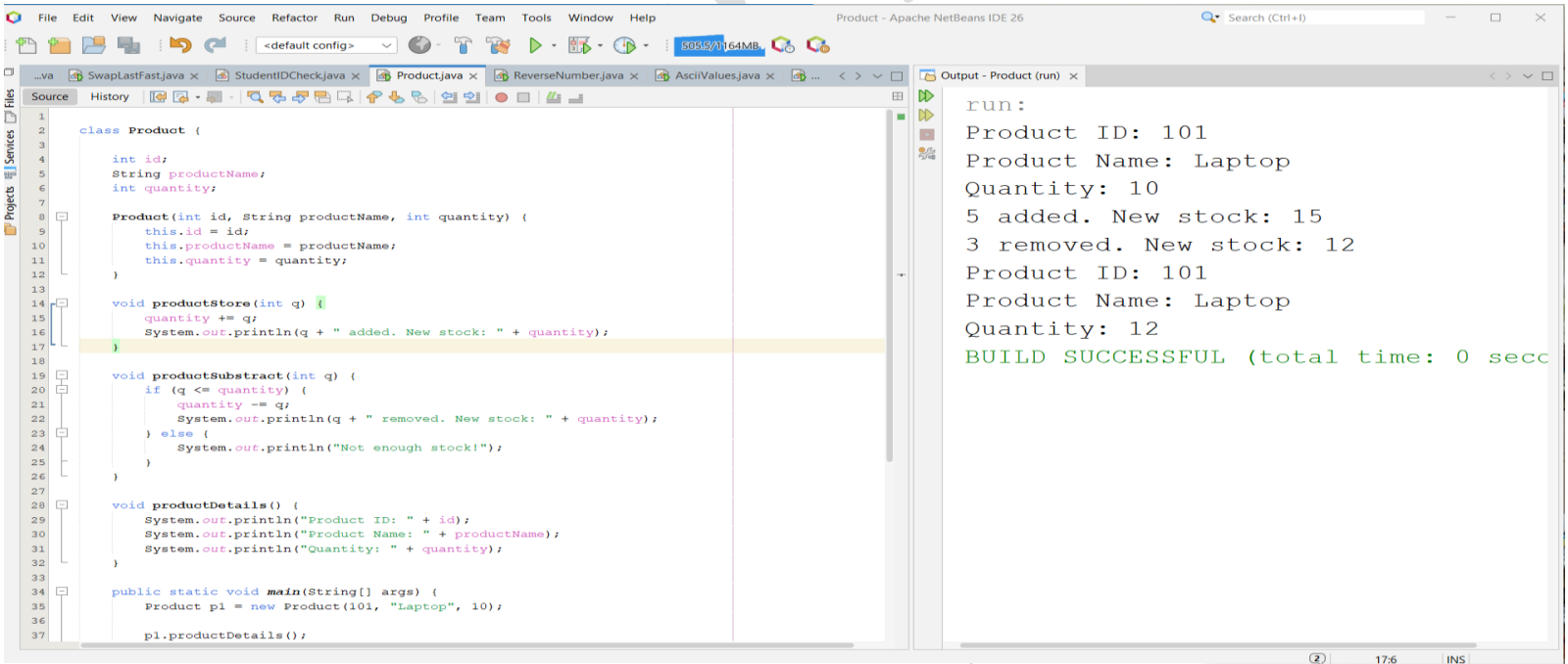
3 .Print the Reverse Number :



```java
import java.util.Scanner;

public class ReverseNumber {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n = sc.nextInt();

        int reverse = 0;
        int temp = n;

        while (temp != 0) {
            int digit = temp % 10;
            reverse = reverse * 10 + digit;
            temp /= 10;
        }

        System.out.println("Reverse = " + reverse);
```

```
run:
Enter a number: 1234
Reverse = 4321
BUILD SUCCESSFUL (total time: 7 seconds)
```

4. Print the Product :

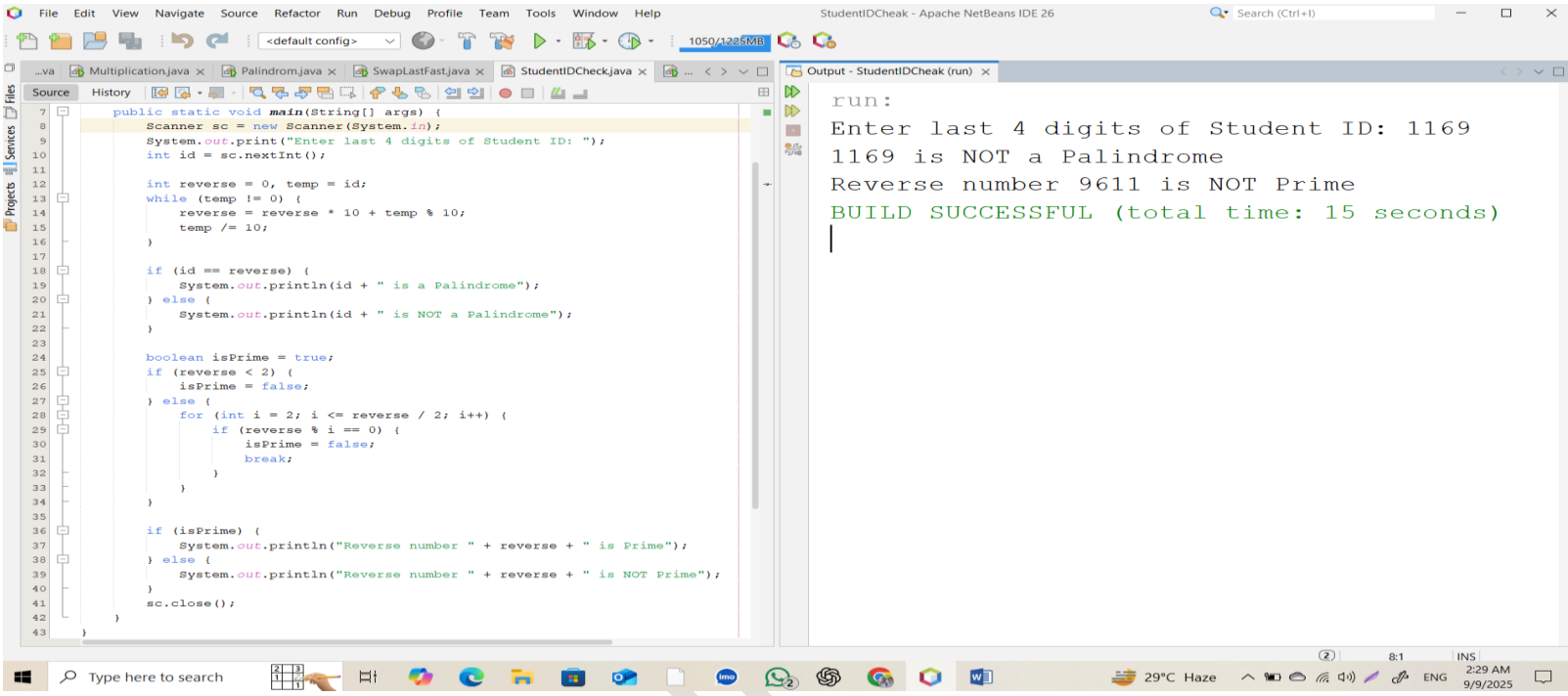


```java
class Product {

    int id;
    String productName;
    int quantity;

    Product(int id, String productName, int quantity) {
        this.id = id;
        this.productName = productName;
        this.quantity = quantity;
    }

    void productStore(int q) {
        quantity += q;
        System.out.println(q + " added. New stock: " + quantity);
    }

    void productSubstract(int q) {
        if (q <= quantity) {
            quantity -= q;
            System.out.println(q + " removed. New stock: " + quantity);
        } else {
            System.out.println("Not enough stock!");
        }
    }

    void productDetails() {
        System.out.println("Product ID: " + id);
        System.out.println("Product Name: " + productName);
        System.out.println("Quantity: " + quantity);
    }

    public static void main(String[] args) {
        Product p1 = new Product(101, "Laptop", 10);

        p1.productDetails();
```

```
run:
Product ID: 101
Product Name: Laptop
Quantity: 10
5 added. New stock: 15
3 removed. New stock: 12
Product ID: 101
Product Name: Laptop
Quantity: 12
BUILD SUCCESSFUL (total time: 0 secc
```

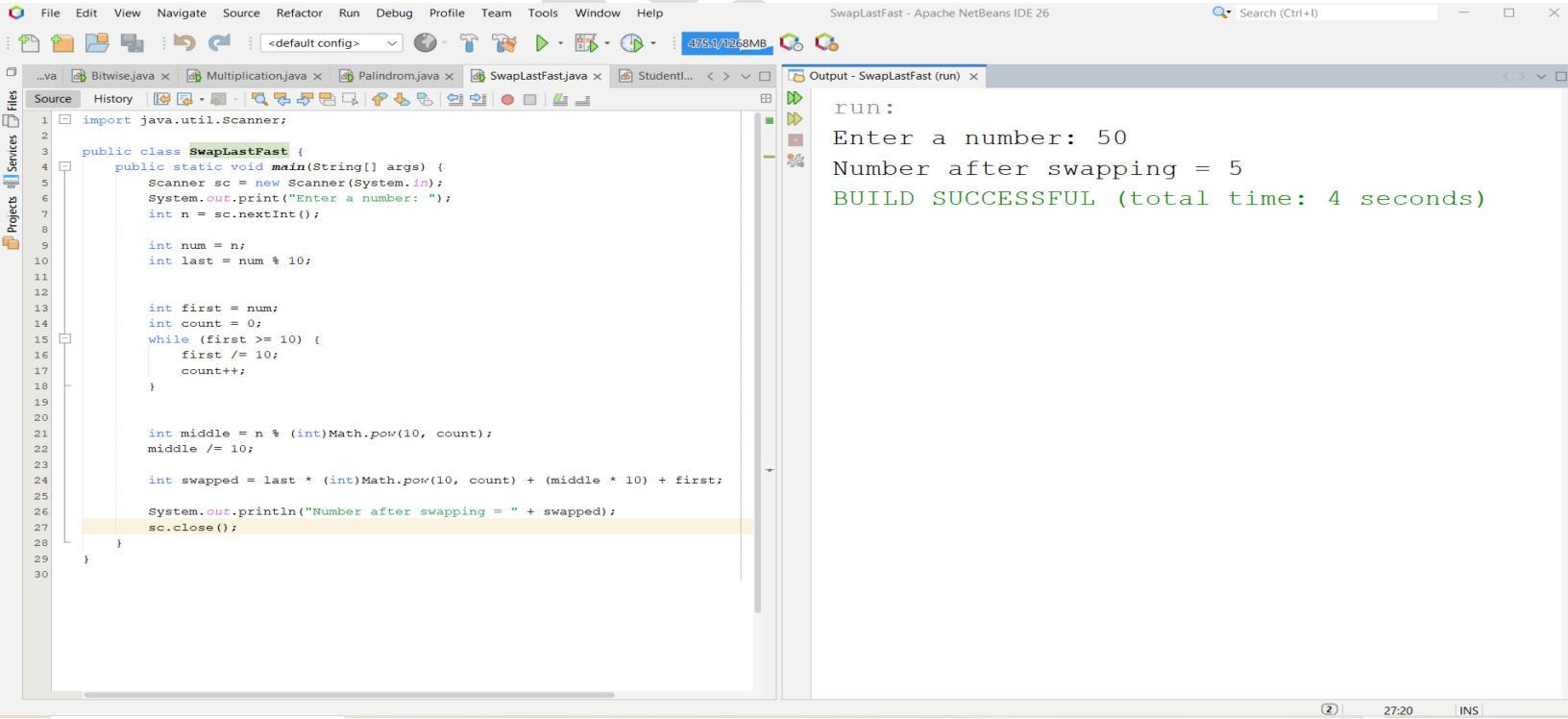
```java
void productDetails() {
    System.out.println("Product ID: " + id);
    System.out.println("Product Name: " + productName);
    System.out.println("Quantity: " + quantity);
}

public static void main(String[] args) {
    Product p1 = new Product(101, "Laptop", 10);

    p1.productDetails();
    p1.productStore(5);
    p1.productSubstract(3);
    p1.productDetails();
}
```

5.Print the StudentIDCheak



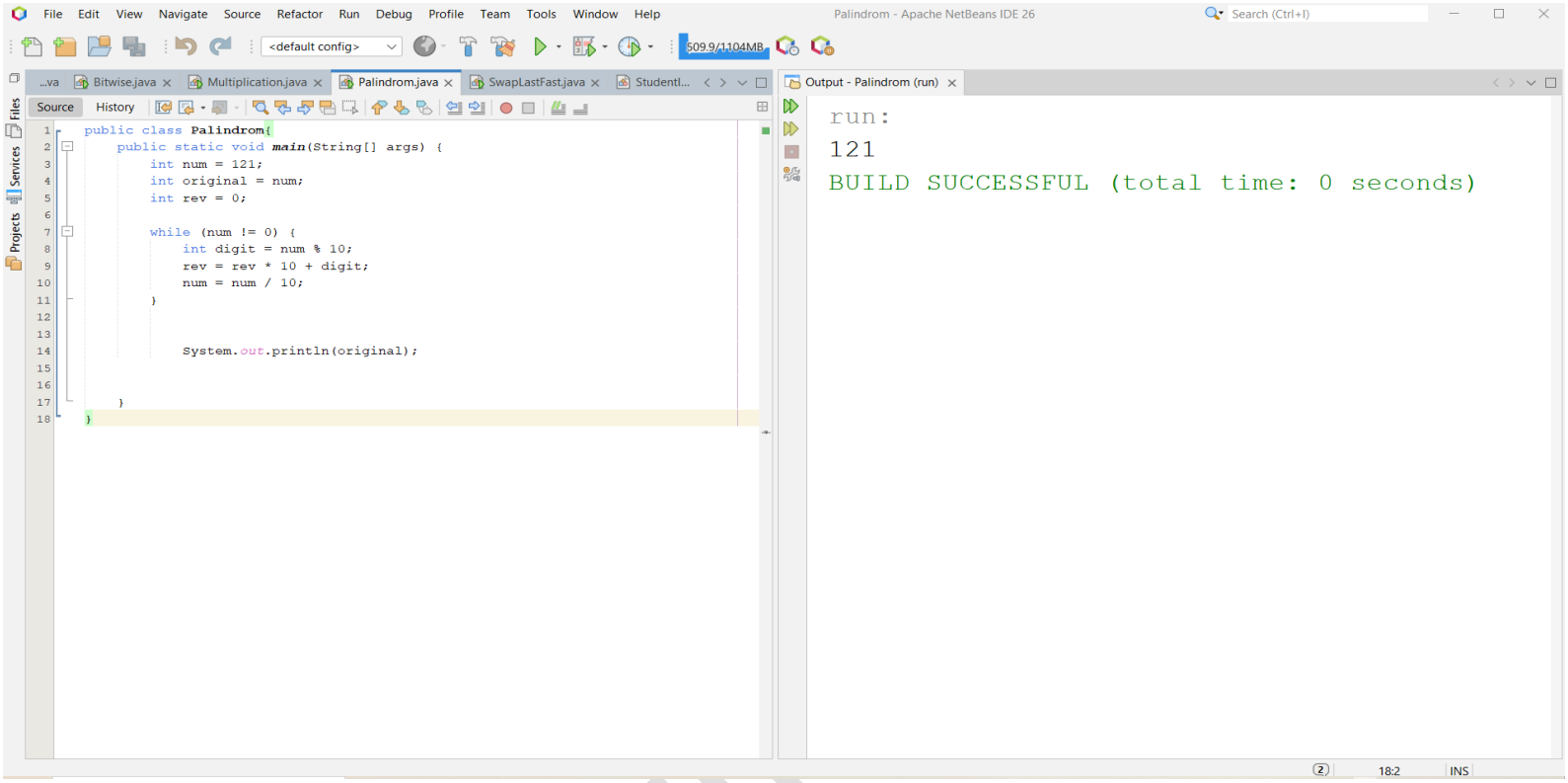
```
run:
Enter last 4 digits of Student ID: 1169
1169 is NOT a Palindrome
Reverse number 9611 is NOT Prime
BUILD SUCCESSFUL (total time: 15 seconds)
```

6.Print the SwapLastFast



```
run:
Enter a number: 50
Number after swapping = 5
BUILD SUCCESSFUL (total time: 4 seconds)
```

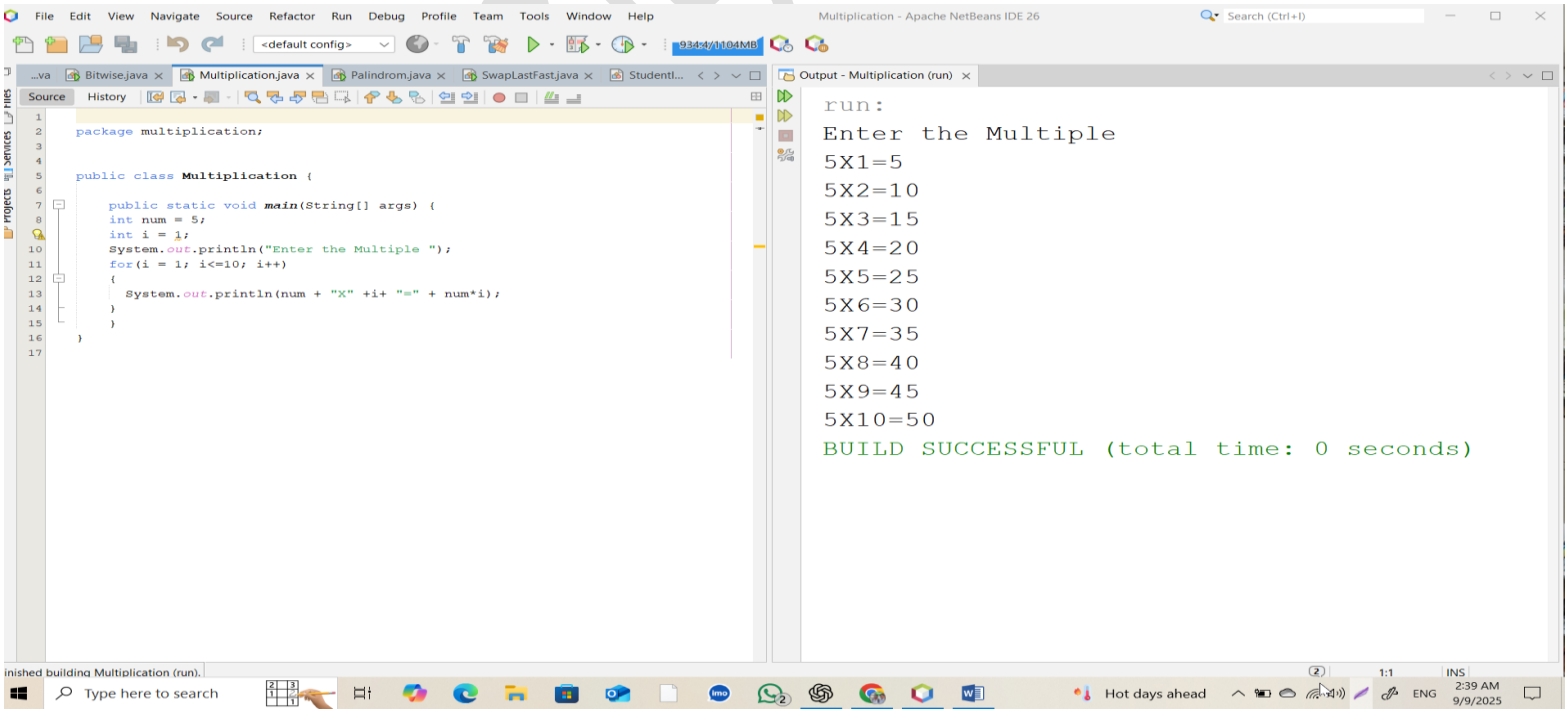
7.Print the Palindrome



```
public class Palindrom{
    public static void main(String[] args) {
        int num = 121;
        int original = num;
        int rev = 0;

        while (num != 0) {
            int digit = num % 10;
            rev = rev * 10 + digit;
            num = num / 10;
        }


            System.out.println(original);


    }
}
```

```
run:
121
BUILD SUCCESSFUL (total time: 0 seconds)
```

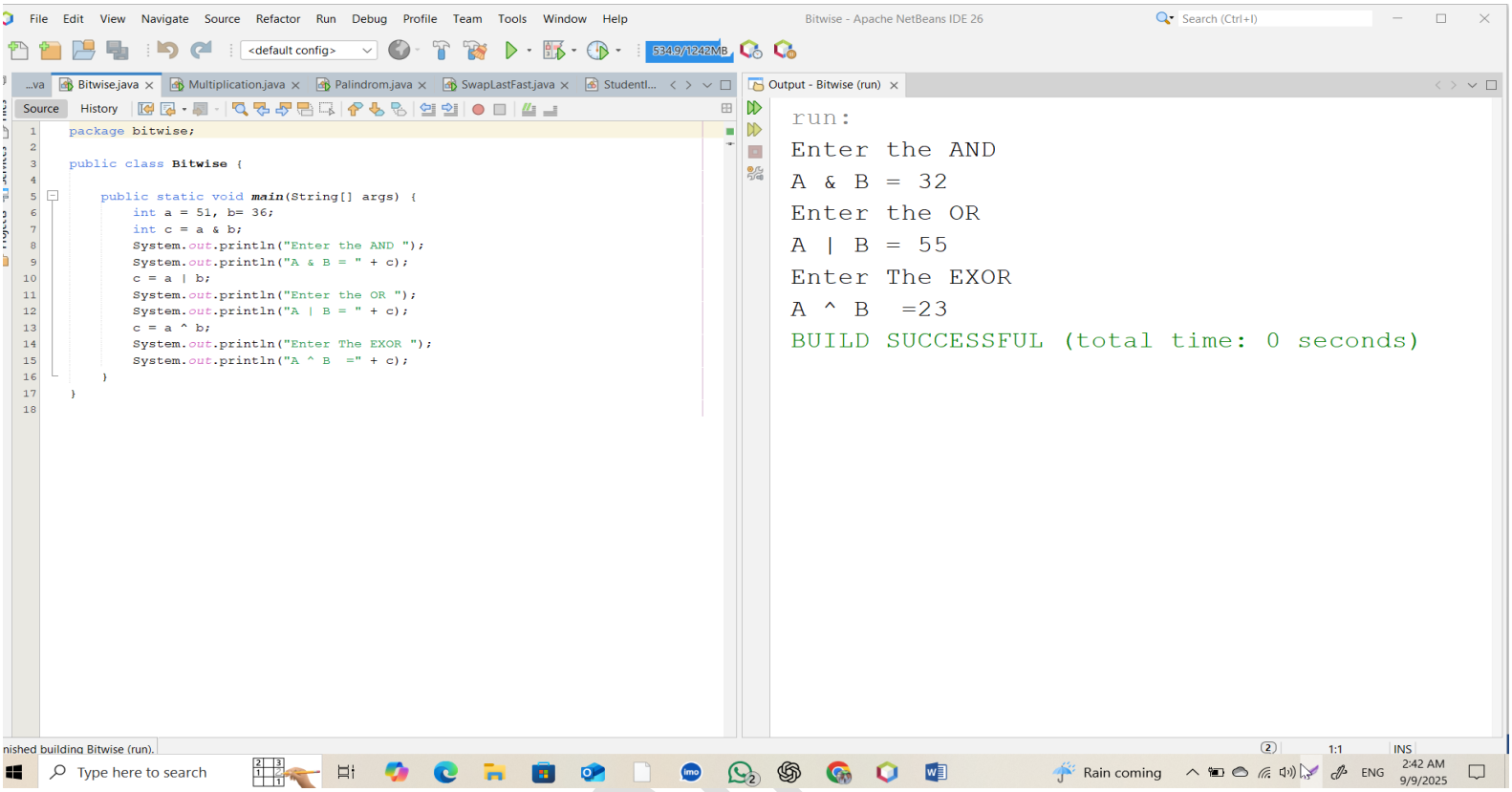
8. Print the Multiplication



```
package multiplication;


public class Multiplication {

    public static void main(String[] args) {
    int num = 5;
    int i = 1;
    System.out.println("Enter the Multiple ");
    for(i = 1; i<=10; i++)
    {
      System.out.println(num + "X" +i+ "=" + num*i);
    }
    }
}
```

```
run:
Enter the Multiple
5X1=5
5X2=10
5X3=15
5X4=20
5X5=25
5X6=30
5X7=35
5X8=40
5X9=45
5X10=50
BUILD SUCCESSFUL (total time: 0 seconds)
```

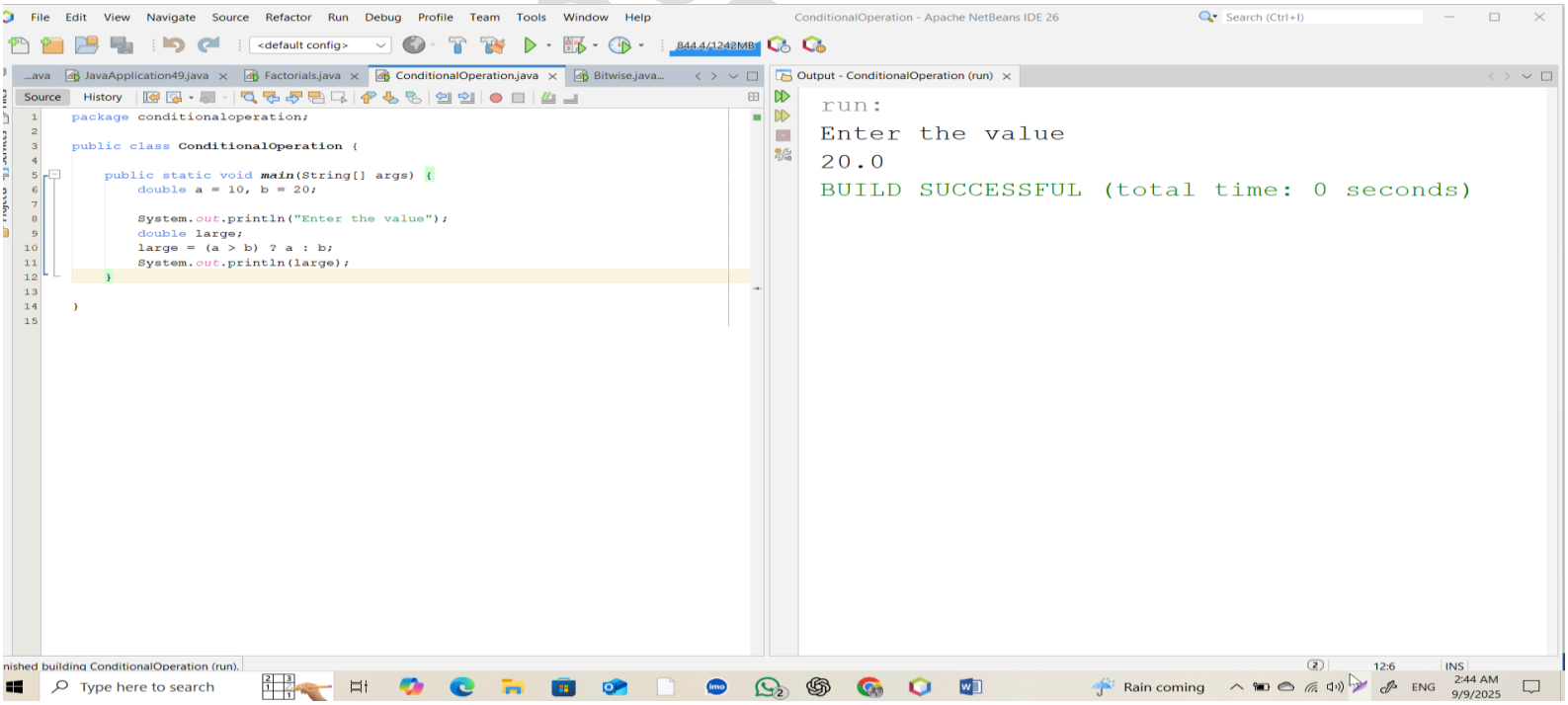
9. Print the Bitwise



```
package bitwise;

public class Bitwise {

    public static void main(String[] args) {
        int a = 51, b= 36;
        int c = a & b;
        System.out.println("Enter the AND ");
        System.out.println("A & B = " + c);
        c = a | b;
        System.out.println("Enter the OR ");
        System.out.println("A | B = " + c);
        c = a ^ b;
        System.out.println("Enter The EXOR ");
        System.out.println("A ^ B  =" + c);
    }
}
```

```
run:
Enter the AND
A & B = 32
Enter the OR
A | B = 55
Enter The EXOR
A ^ B  =23
BUILD SUCCESSFUL (total time: 0 seconds)
```

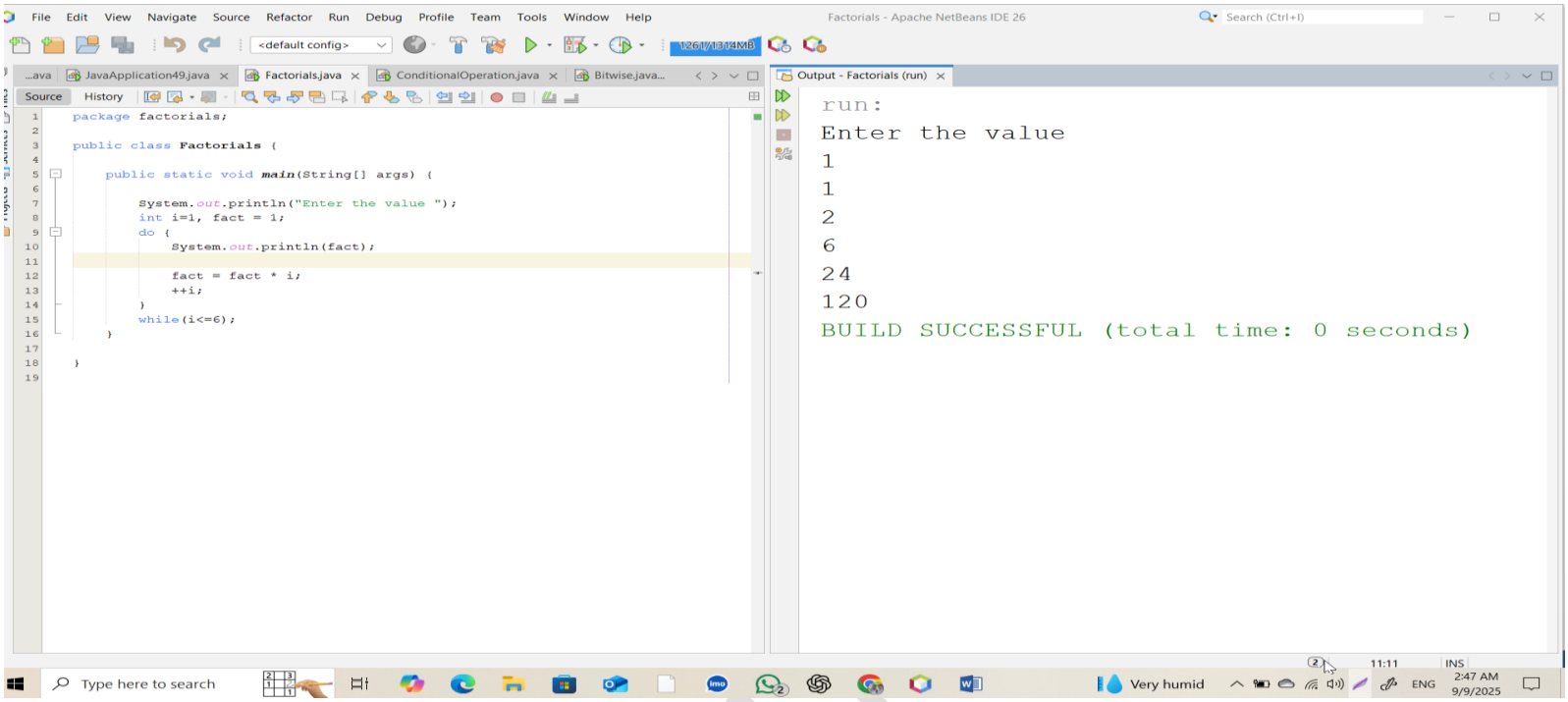
10. Print the ConditionalOperation



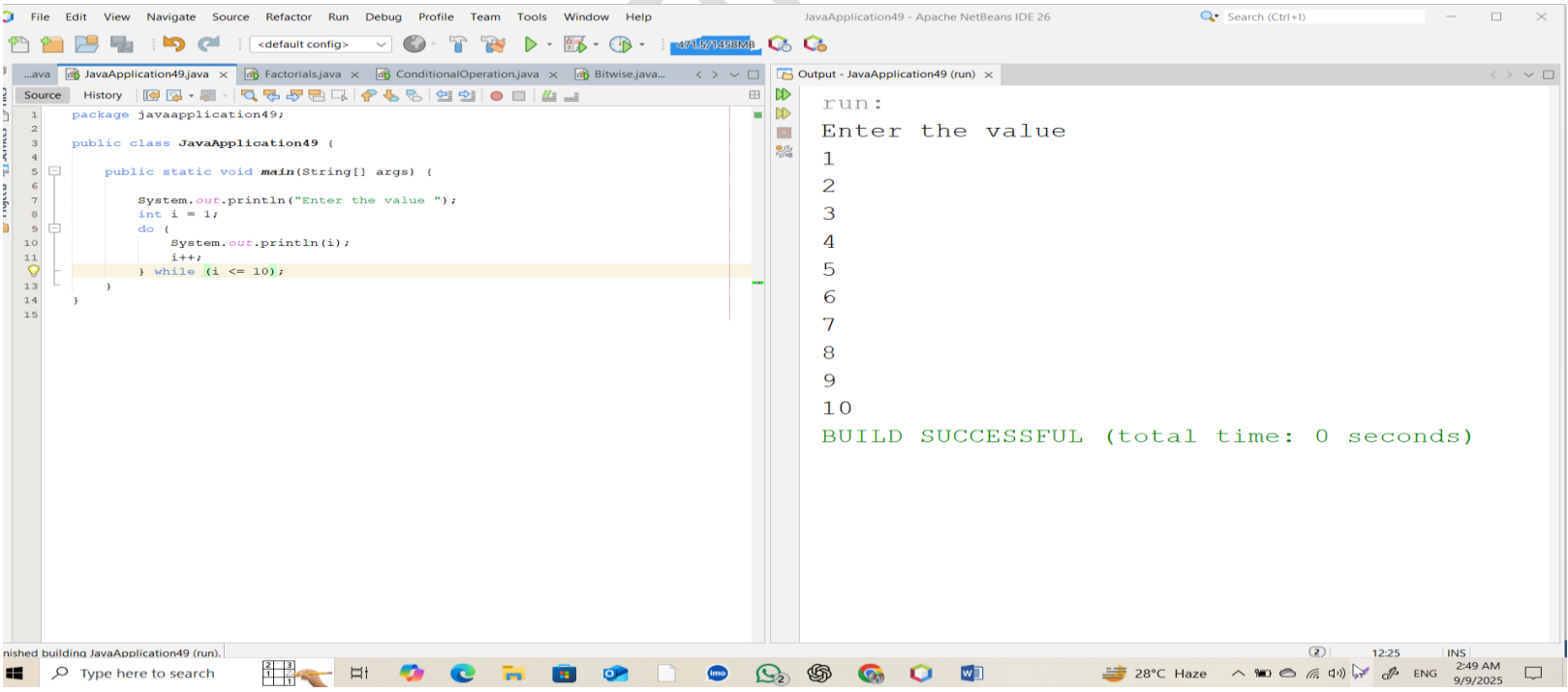
```
package conditionaloperation;

public class ConditionalOperation {

    public static void main(String[] args) {
        double a = 10, b = 20;

        System.out.println("Enter the value");
        double large;
        large = (a > b) ? a : b;
        System.out.println(large);
    }

}
```

```
run:
Enter the value
20.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

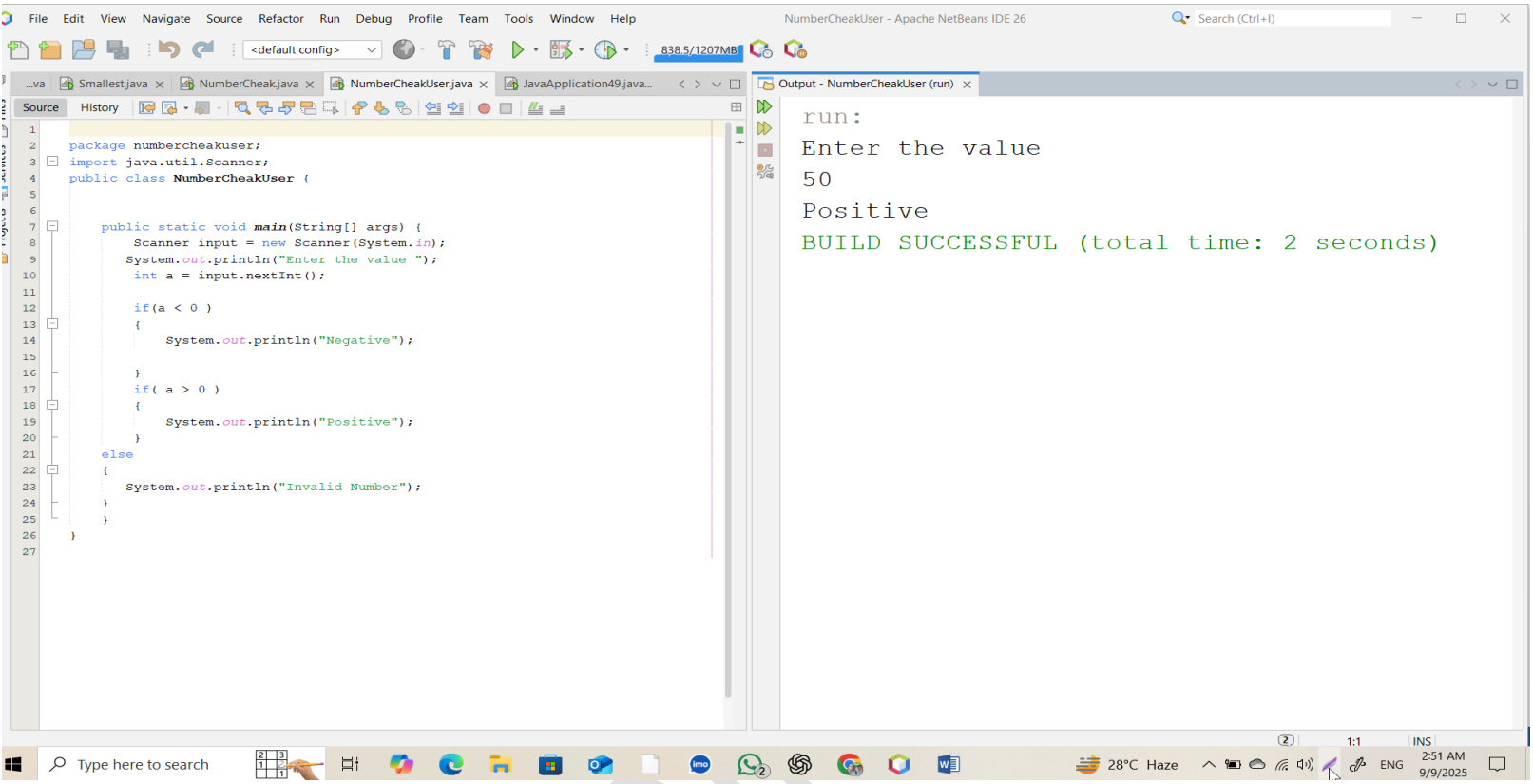
11. Print the Factorials



12. Print the Number

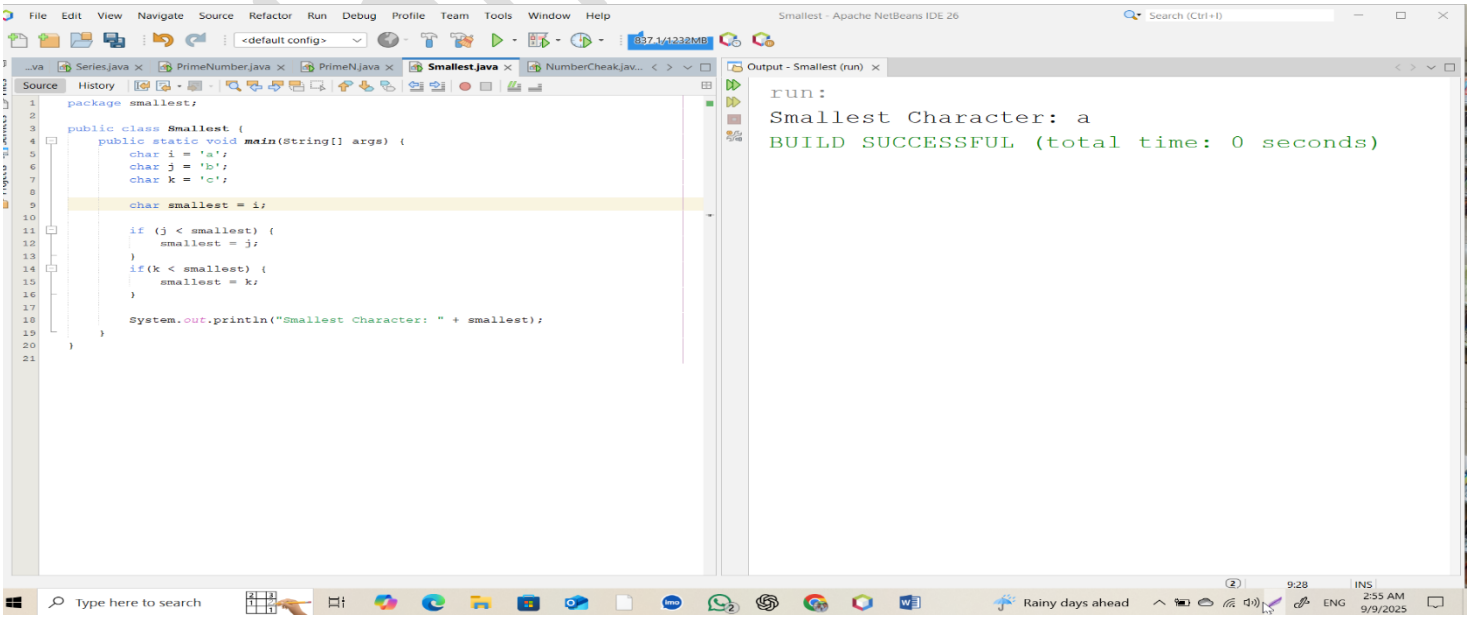


13. Print the Nunber cheak



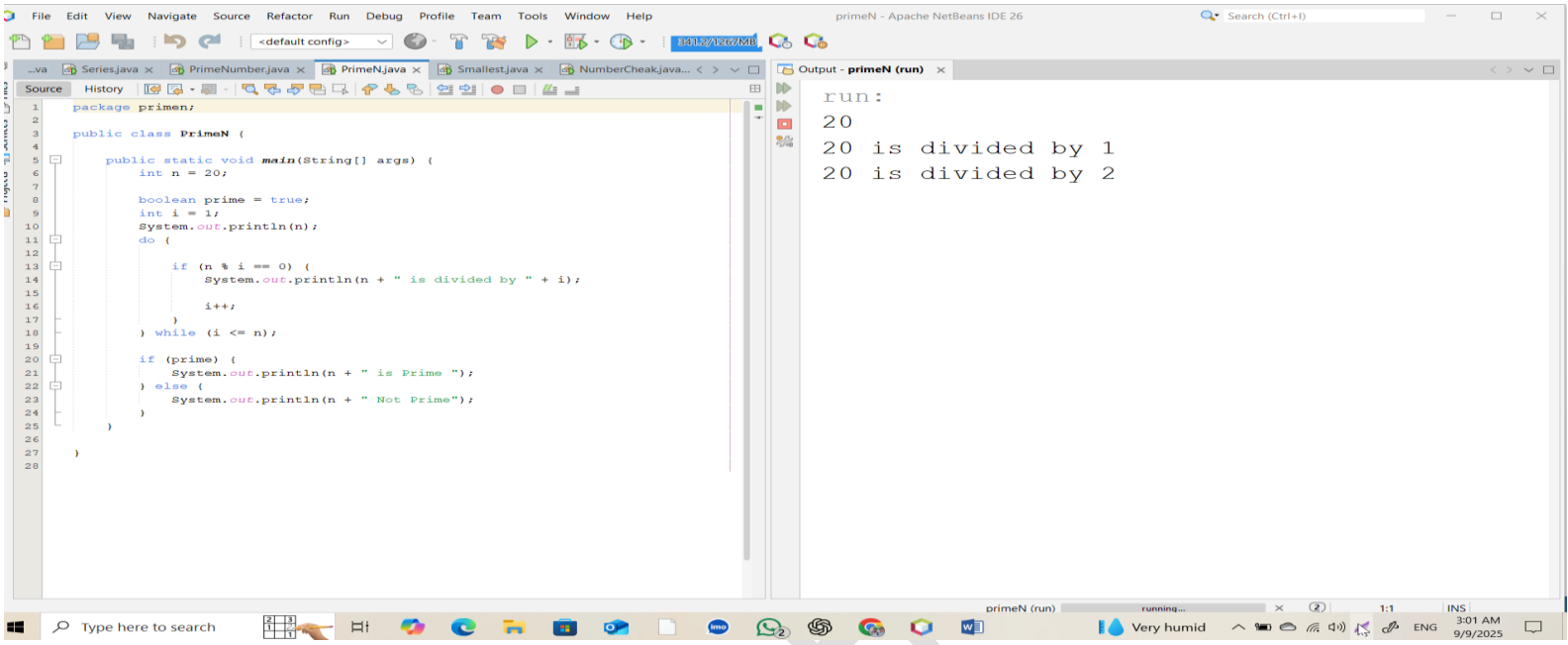

```java
package numbercheakuser;
import java.util.Scanner;
public class NumberCheakUser {


    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
       System.out.println("Enter the value ");
        int a = input.nextInt();

        if(a < 0 )
        {
            System.out.println("Negative");

        }
        if( a > 0 )
        {
            System.out.println("Positive");
        }
    else
    {
       System.out.println("Invalid Number");
    }
    }
}
```

14.Print the Cheak Smallest




```java
package smallest;

public class Smallest {
    public static void main(String[] args) {
        char i = 'a';
        char j = 'b';
        char k = 'c';

        char smallest = i;

        if (j < smallest) {
            smallest = j;
        }
        if(k < smallest) {
            smallest = k;
        }

        System.out.println("Smallest Character: " + smallest);
    }
}
```

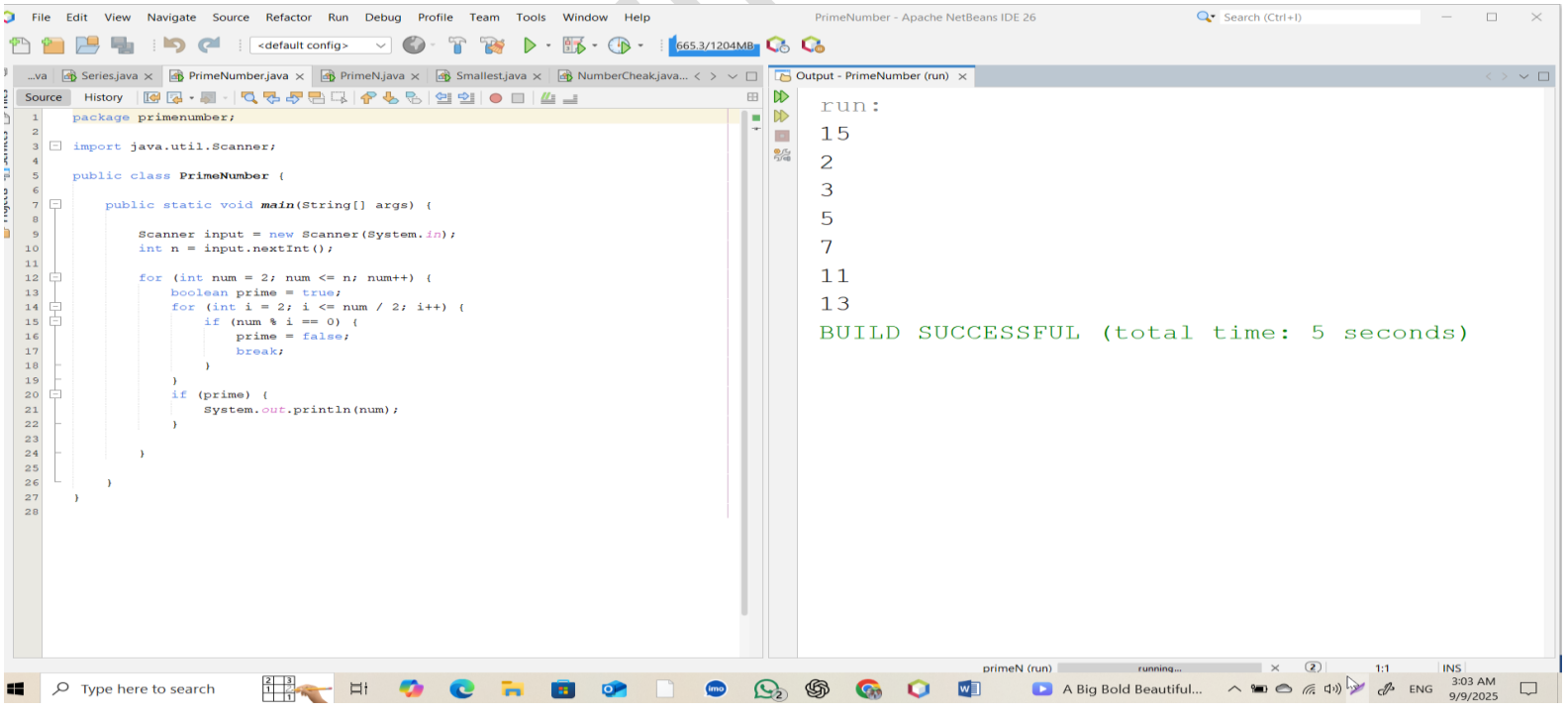
15. Print the Prime Number



```
package primen;

public class PrimeN {

    public static void main(String[] args) {
        int n = 20;

        boolean prime = true;
        int i = 1;
        System.out.println(n);
        do {

            if (n % i == 0) {
                System.out.println(n + " is divided by " + i);

                i++;
            }
        } while (i <= n);

        if (prime) {
            System.out.println(n + " is Prime ");
        } else {
            System.out.println(n + " Not Prime");
        }
    }

}
```

```
run:
20
20 is divided by 1
20 is divided by 2
```

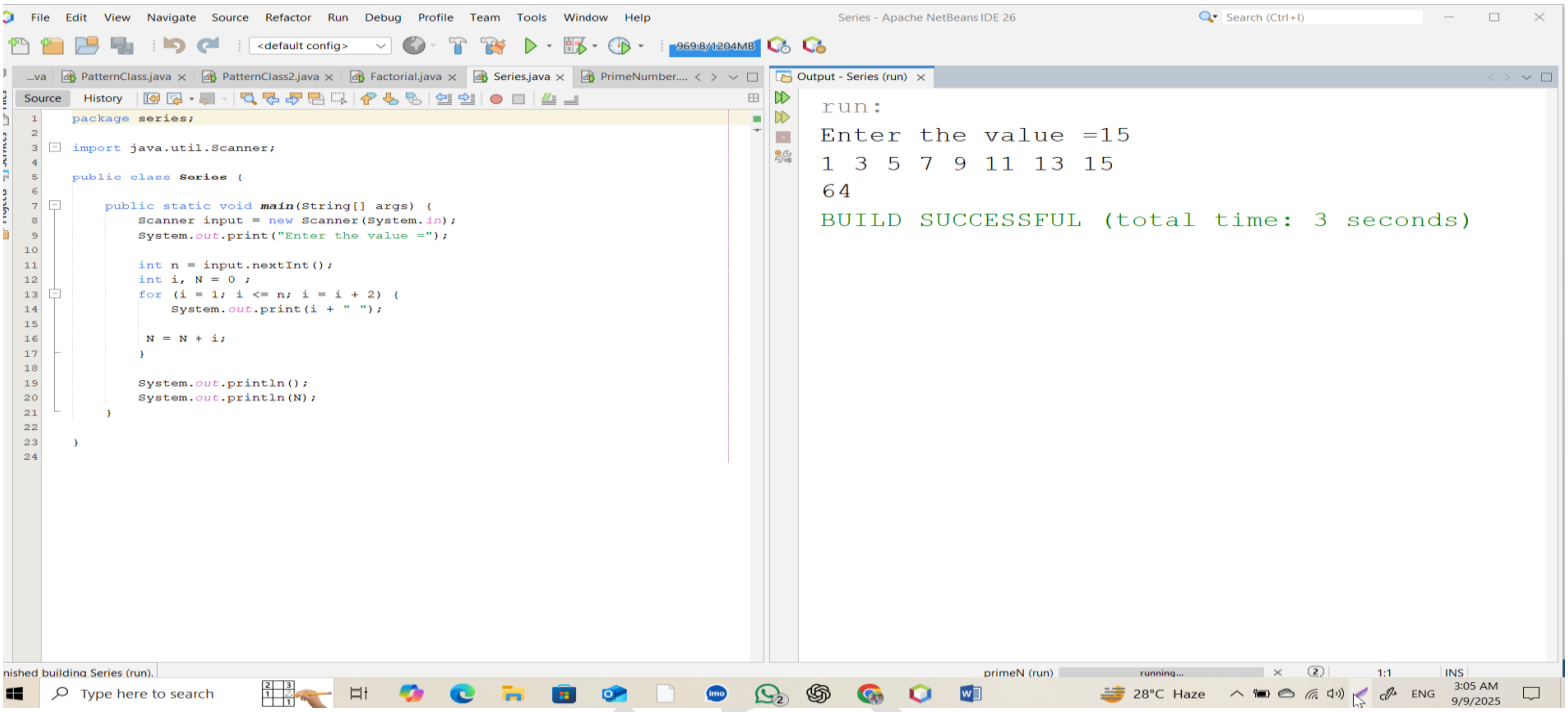
16. Print the Prime Number from user



```
package primenumber;

import java.util.Scanner;

public class PrimeNumber {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        int n = input.nextInt();

        for (int num = 2; num <= n; num++) {
            boolean prime = true;
            for (int i = 2; i <= num / 2; i++) {
                if (num % i == 0) {
                    prime = false;
                    break;
                }
            }
            if (prime) {
                System.out.println(num);
            }

        }

    }
}
```

```
run:
15
2
3
5
7
11
13
BUILD SUCCESSFUL (total time: 5 seconds)
```

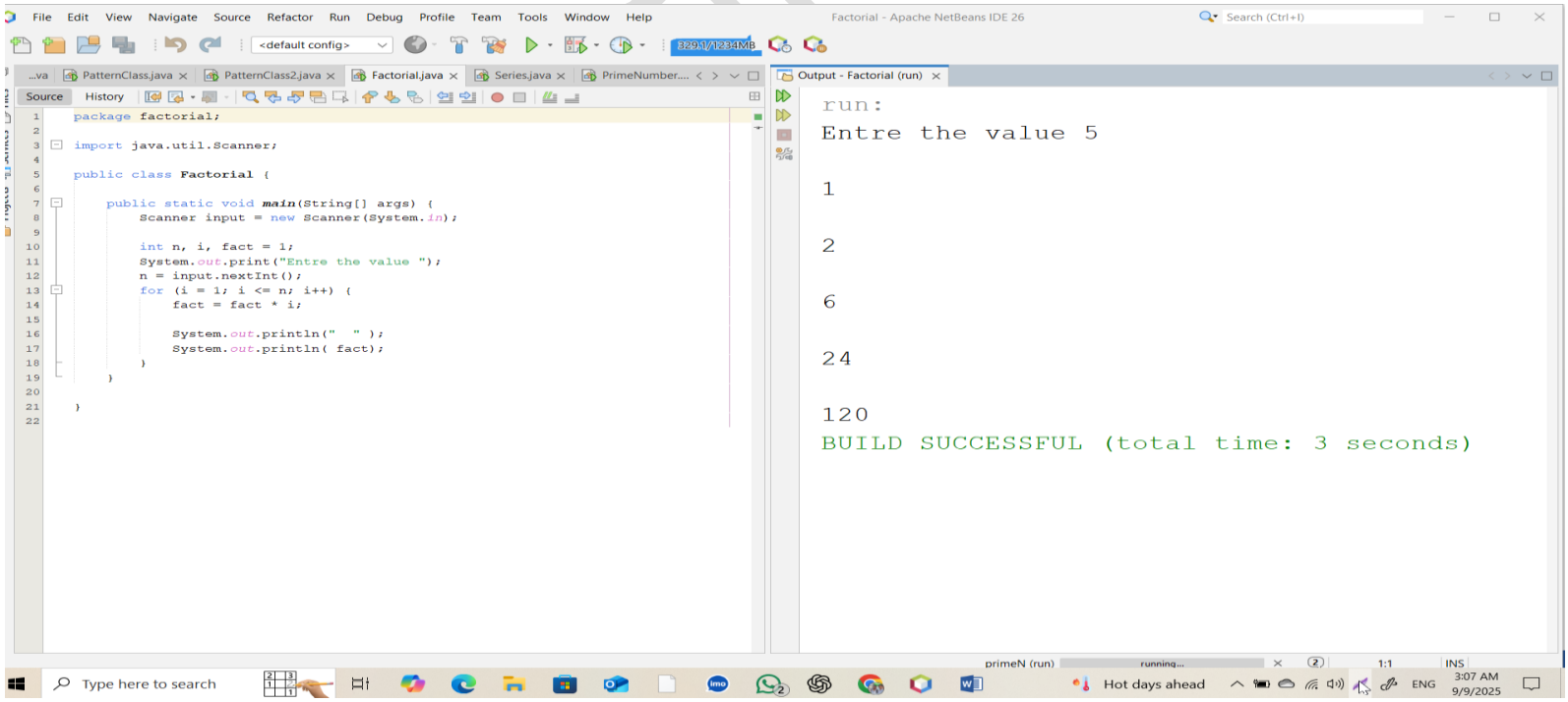
17. Print the Series



```
package series;

import java.util.Scanner;

public class Series {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the value =");

        int n = input.nextInt();
        int i, N = 0 ;
        for (i = 1; i <= n; i = i + 2) {
            System.out.print(i + " ");

         N = N + i;
        }

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

}
```

```
run:
Enter the value =15
1 3 5 7 9 11 13 15
64
BUILD SUCCESSFUL (total time: 3 seconds)
```

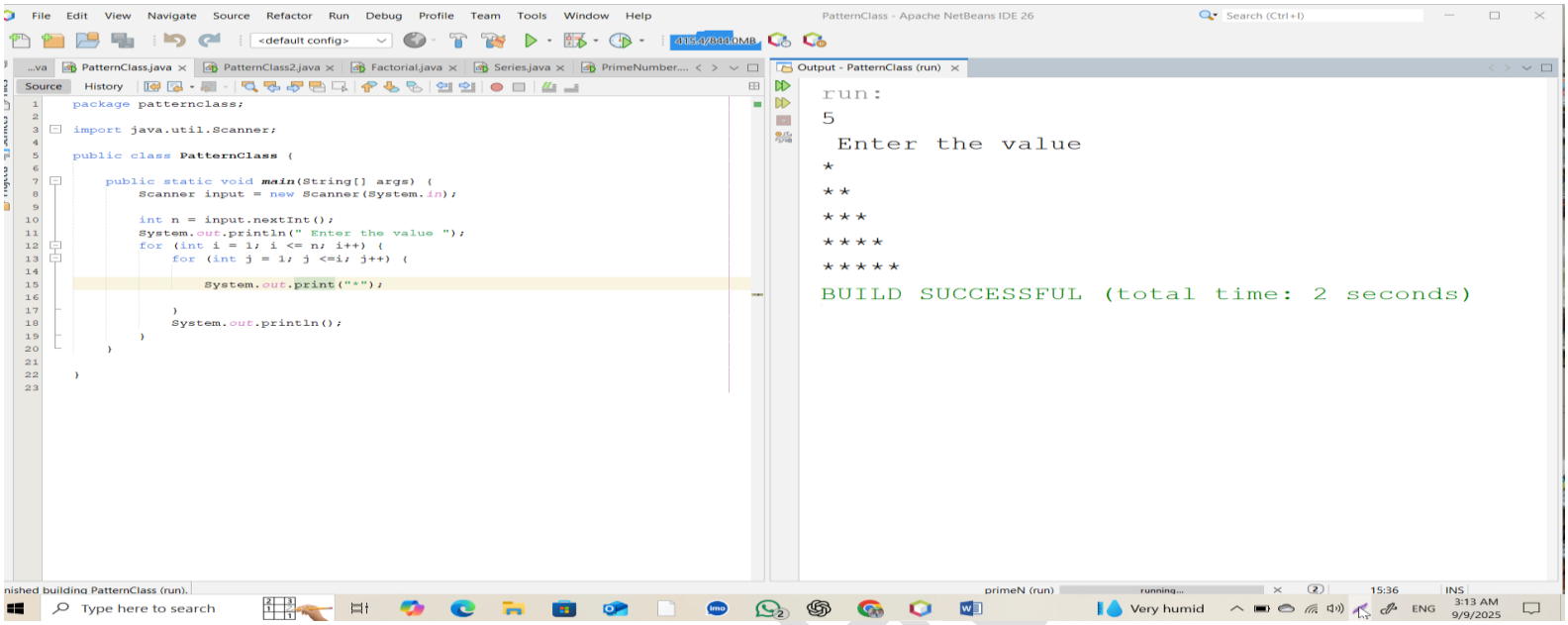
18. Print the Factorials



```
package factorial;

import java.util.Scanner;

public class Factorial {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int n, i, fact = 1;
        System.out.print("Entre the value ");
        n = input.nextInt();
        for (i = 1; i <= n; i++) {
            fact = fact * i;

            System.out.println("  " );
            System.out.println( fact);
        }
    }

}
```

```
run:
Entre the value 5

1

2

6

24

120
BUILD SUCCESSFUL (total time: 3 seconds)
```

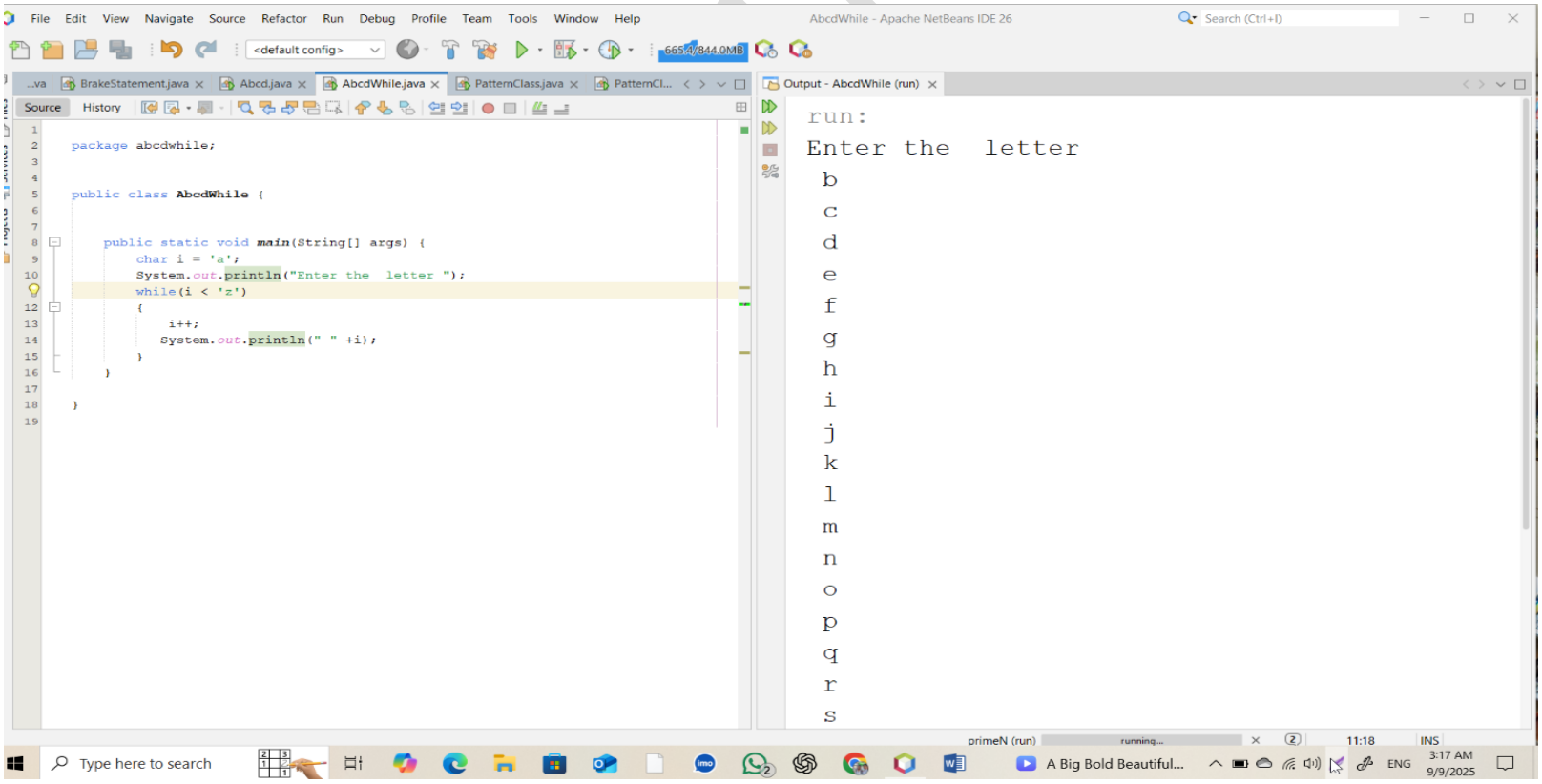
19.Print the Pattern



```
package patternclass;

import java.util.Scanner;

public class PatternClass {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int n = input.nextInt();
        System.out.println(" Enter the value ");
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <=i; j++) {

                System.out.print("*");

            }
            System.out.println();
        }
    }

}
```

```
run:
5
 Enter the value
*
**
***
****
*****
BUILD SUCCESSFUL (total time: 2 seconds)
```

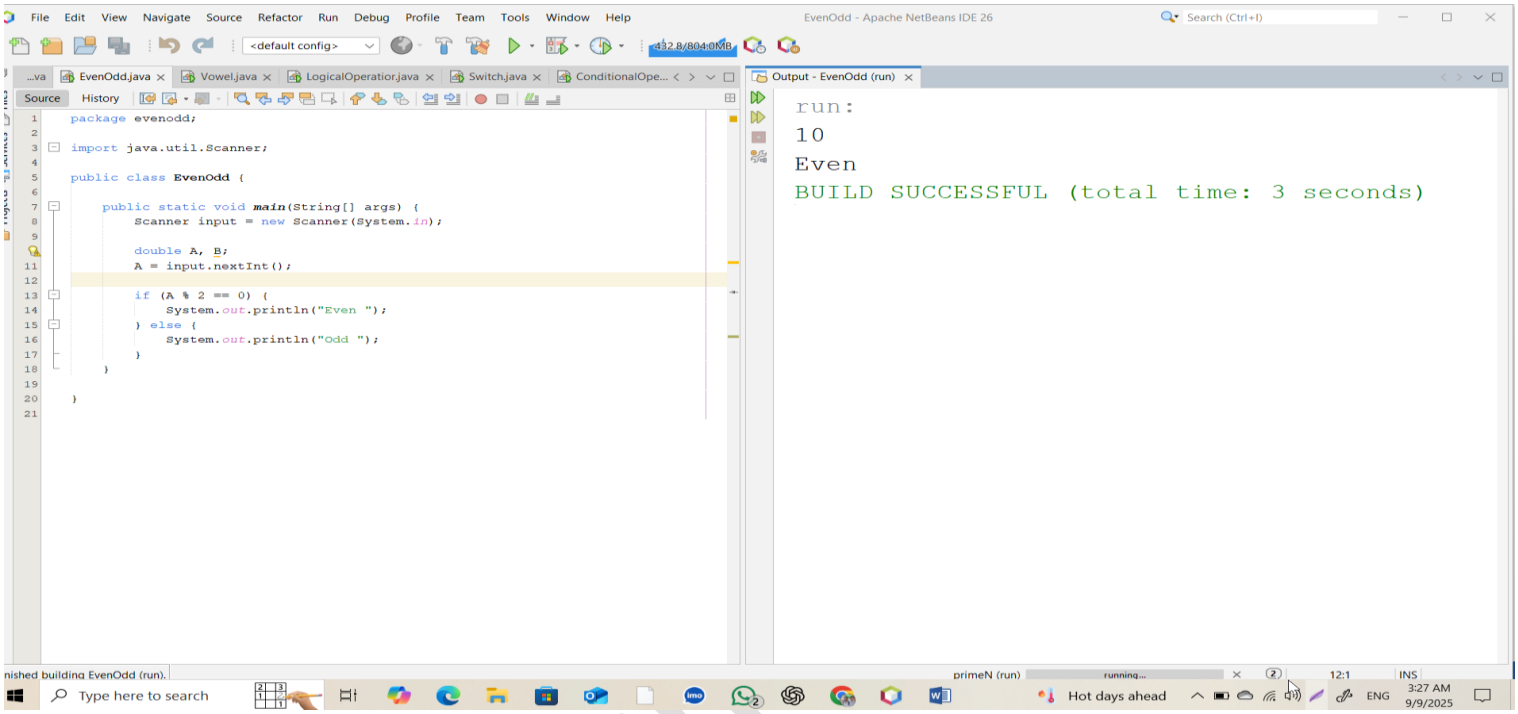
20. Print the Small latter



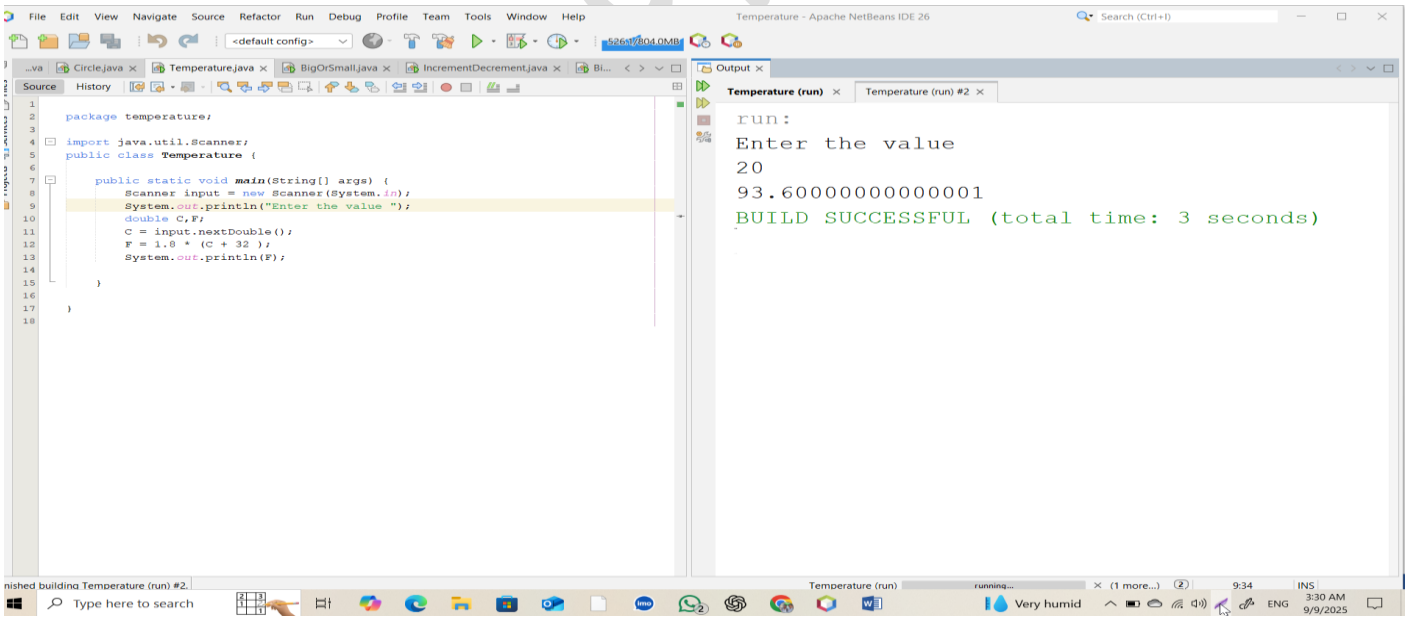
```
package abcdwhile;


public class AbcdWhile {


    public static void main(String[] args) {
        char i = 'a';
        System.out.println("Enter the  letter ");
        while(i < 'z')
        {
            i++;
        System.out.println(" " +i);
        }
    }

}
```

```
run:
Enter the  letter
 b
 c
 d
 e
 f
 g
 h
 i
 j
 k
 l
 m
 n
 o
 p
 q
 r
 s
```

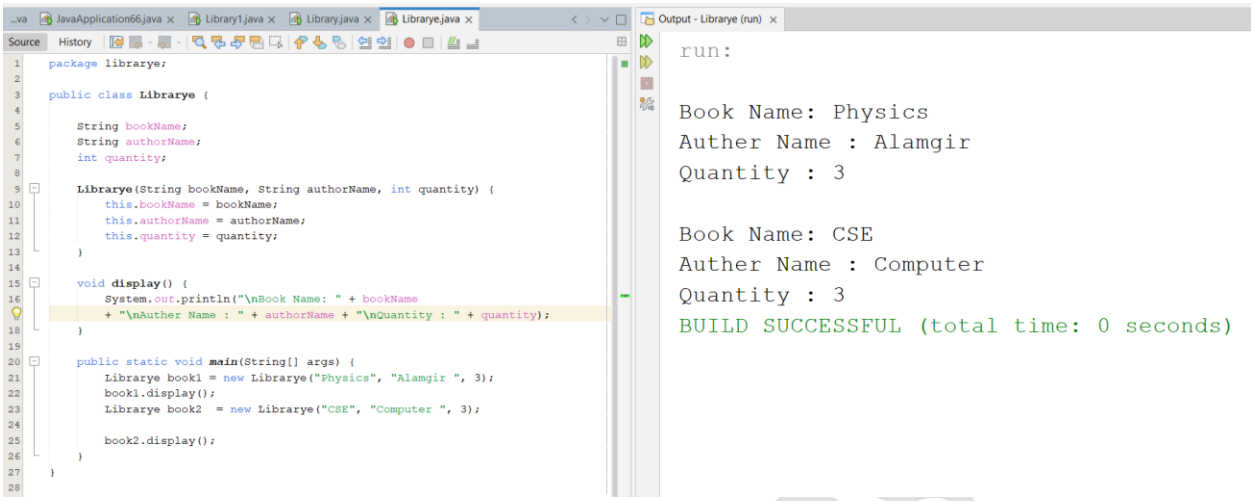
21.Print the Even or Odd



22. Print the Temperature



23. Print the Librarye



```java
package librarye;

public class Librarye {

    String bookName;
    String authorName;
    int quantity;

    Librarye(String bookName, String authorName, int quantity) {
        this.bookName = bookName;
        this.authorName = authorName;
        this.quantity = quantity;
    }

    void display() {
        System.out.println("\nBook Name: " + bookName
        + "\nAuther Name : " + authorName + "\nQuantity : " + quantity);
    }

    public static void main(String[] args) {
        Librarye book1 = new Librarye("Physics", "Alamgir ", 3);
        book1.display();
        Librarye book2  = new Librarye("CSE", "Computer ", 3);

        book2.display();
    }
}
```

```
run:

Book Name: Physics
Auther Name : Alamgir
Quantity : 3

Book Name: CSE
Auther Name : Computer
Quantity : 3
BUILD SUCCESSFUL (total time: 0 seconds)
```


Name : Md Alamgir Hosen

ID: 02724205101169

Batch : 65th

Section: C

Depertment of CSE, City University

Github Link : https://github.com/alamgirsu/alamgirsu2526.git