Practical File

OOPS with JAVA

Code- ARI256

2023-24

Submitted by:

Name-> Shubham Dev

Branch-> IIOT-B1

Enrolment no-> 01919011722

Submitted To:

Dr. Neeta Singh

Professor

USAR, GGSIPU

University School of Automation and Robotics

East Campus, GGSIP University

Surajmal Vihar, New Delhi - 110092

Name :Shubham Dev Enrolment No : 01919011722 Batch : IIOT-B1

INDEX

S.NO.	Date	Title	Teacher's Sign

Name :Shubham Dev Enrolment No : 01919011722 Batch : IIOT-B1

Lab – 1

Aim– (A) Write a Program to print “Hello World” using command prompt.

Code–

```
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello World");
        }
}
```

Output–

```
C:\Users\devsh>cd Desktop

C:\Users\devsh\Desktop>javac Main.java

C:\Users\devsh\Desktop>java Main.java
Hello World
```

(B) Write a Program to calculate the area of rectangle using command prompt.

Code-

```
public class AreaOfRectangle{
        public static void main(String args[]){
                int length = 25;
                int breadth =12;
                int area = length * breadth;
                System.out.println("Area of Rectangle is: " + area);
                }
        }
```

Output-

```
C:\Users\devsh\Desktop>javac AreaOfRectangle.java

C:\Users\devsh\Desktop>java AreaOfRectangle.java
Area of Rectangle is: 300

C:\Users\devsh\Desktop>
```

Name :Shubham Dev Enrolment No : 01919011722 Batch : IIOT-B1

Lab – 2

Aim– (A) Write a Program to find whether the number is prime number or not using 'for loop'.

Code–

Main.java

```java
public class Main {
    public static void main(String[] args) {
        int num = 29;
        boolean flag = false;
        for (int i = 2; i <= num / 2; ++i) {
        // condition for non-prime number
        if (num % i == 0) {
            flag = true;
            break;
        }
      }
      if (!flag) System.out.println(num + " is a prime number.");
      else System.out.println(num + " is not a prime number.");
    }
}
```

Output–

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "
29 is a prime number.
```

Name :Shubham Dev Enrolment No : 01919011722 Batch : IIOT-B1

(B) Write a Program to find whether the number is prime number or not using 'while loop'.

Code-

Main.java

```java
public class Main {
  public static void main(String[] args) {
    int num = 33, i = 2;
    boolean flag = false;
    while (i <= num / 2) {
      // condition for non-prime number
      if (num % i == 0) {
        flag = true;
        break;
      }
      ++i;
    }
    if (!flag) System.out.println(num + " is a prime number.");
    else System.out.println(num + " is not a prime number.");
  }
}
```

Output-

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaager

33 is not a prime number.
```

Lab – 3

Aim– Write a program to use the Java Operators

(A) Arithmetic Operators

Code–

```
class Main {
  public static void main(String[] args) {
    // declare variables
    int a = 12, b = 5;
    // addition operator
    System.out.println("a + b = " + (a + b));
    // subtraction operator
    System.out.println("a - b = " + (a - b));
    // multiplication operator
    System.out.println("a * b = " + (a * b));
    // division operator
    System.out.println("a / b = " + (a / b));
    // modulo operator
    System.out.println("a % b =" + (a % b));
  }
}
```

Output–

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagen
a + b = 17
a - b = 7
a * b = 60
a / b = 2
a % b =2
```

(B) Assignment Operators

```
class Main {
  public static void main(String[] args) {
    // create variables
```

```
    int a = 4;
    int var;
    // assign value using =
    var = a;
    System.out.println("var using =: " + var);
    // assign value using =+
    var += a;
    System.out.println("var using +=: " + var);
    // assign value using =*
    var *= a;
    System.out.println("var using *=: " + var);
  }
}
```

Output-

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-java
var using =: 4
var using +=: 8
var using *=: 32

Process finished with exit code 0
```

(C) Relational Operators

```
class Main {
  public static void main(String[] args) {
    // create variables
    int a = 7, b = 11;
    // value of a and b
    System.out.println("a is " + a + " and b is " + b);
    // == operator
    System.out.println(a == b);  // false
    // != operator
    System.out.println(a != b);  // true
    // > operator
    System.out.println(a > b);  // false
    // < operator
    System.out.println(a < b);  // true
    // >= operator
    System.out.println(a >= b);  // false
    // <= operator
    System.out.println(a <= b);  // true
  }
}
```

Output-

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-jav
a is 7 and b is 11
false
true
false
true
false
true

Process finished with exit code 0
```

(D) Java Unary Operators

```
class Main {
  public static void main(String[] args) {
    // declare variables
    int a = 12, b = 12;
    int result1, result2;
    // original value
    System.out.println("Value of a: " + a);
    // increment operator
    result1 = ++a;
    System.out.println("After increment: " + result1);
    System.out.println("Value of b: " + b);
    // decrement operator
    result2 = --b;
    System.out.println("After decrement: " + result2);
  }
}
```

Output-

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-ja
Value of a: 12
After increment: 13
Value of b: 12
After decrement: 11

Process finished with exit code 0
```

Lab – 4

Aim– Write a program to print the Star, Number and Character pattern using Java packages

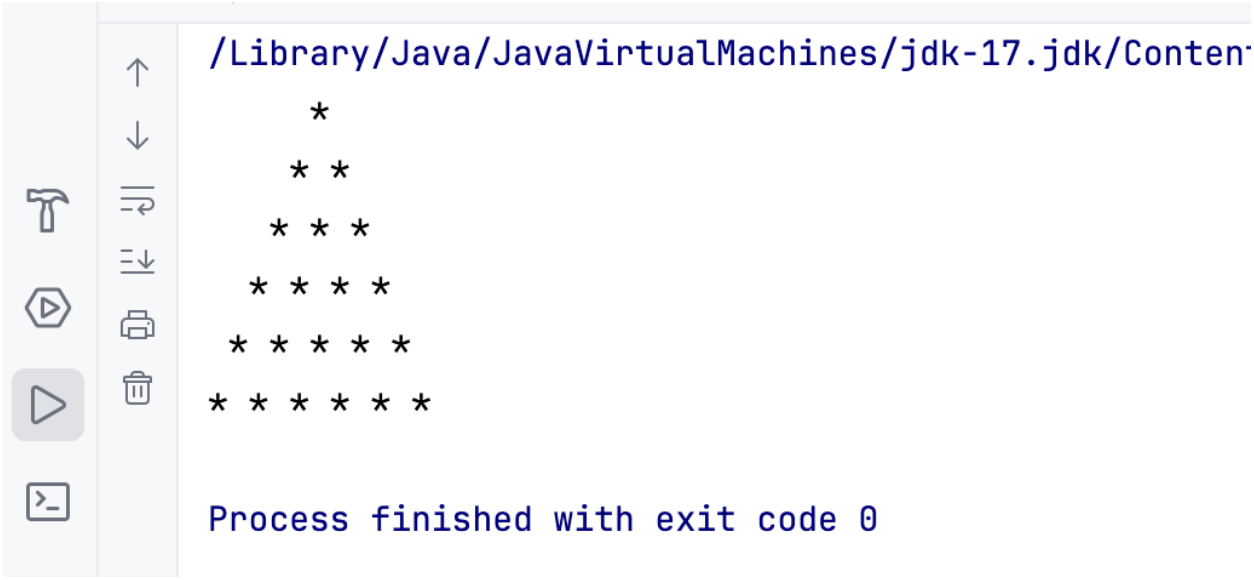
(A) Star Pattern

i) Pyramidal Pattern

Code–

```
public class Main
{
    public static void main(String args[])
    {
        //i for rows and j for columns
        //row denotes the number of rows you want to print
        int i, j, row = 6;
        //Outer loop work for rows
        for (i=0; i<row; i++)
        {
            //inner loop work for space
            for (j=row-i; j>1; j--)
            {
                //prints space between two stars
                System.out.print(" ");
            }
            //inner loop for columns
            for (j=0; j<=i; j++ )
            {
                //prints star
                System.out.print("* ");
            }
            //throws the cursor in a new line after printing
each line
            System.out.println();
        }
    }
}
```

Output-



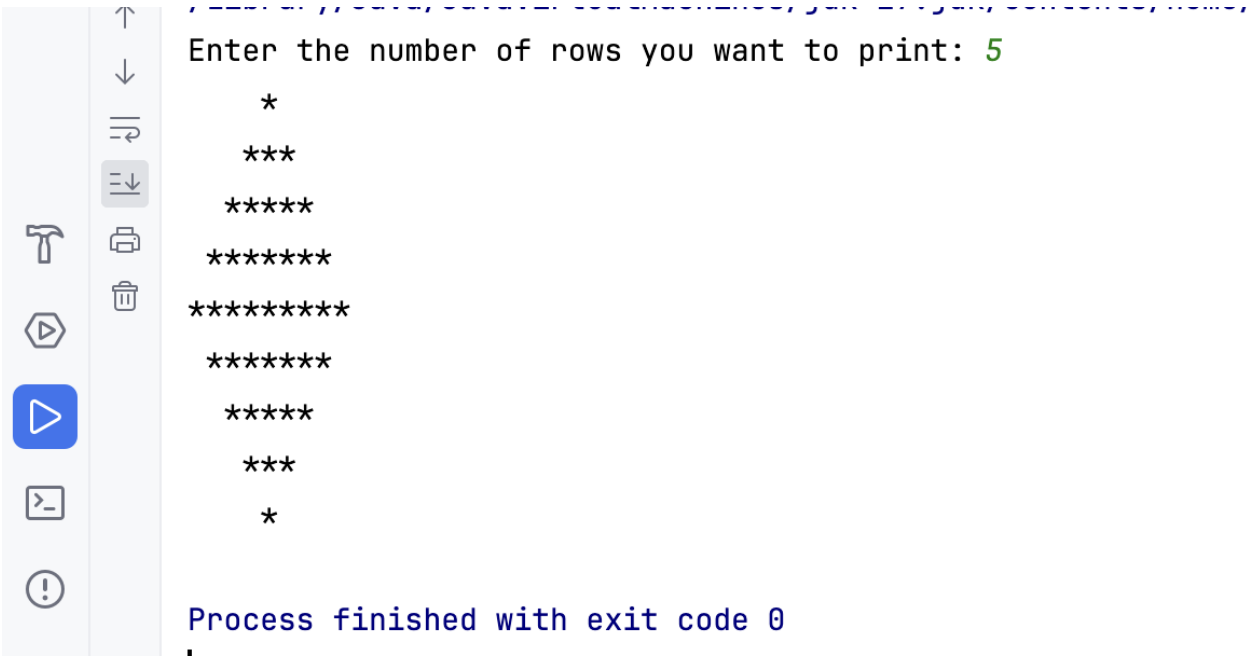
ii) Diamond Shape Pattern

Code-

```
import java.util.Scanner;
public class Experiment_4_B
{
    public static void main(String args[])
    {
        int row, i, j, space = 1;
        System.out.print("Enter the number of rows you want
to print: ");
        Scanner sc = new Scanner(System.in);
        row = sc.nextInt();
        space = row - 1;
        for (j = 1; j<= row; j++)
        {
            for (i = 1; i<= space; i++)
            {
                System.out.print(" ");
            }
            space--;
            for (i = 1; i <= 2 * j - 1; i++)
            {
                System.out.print("*");
            }
            System.out.println("");
        }
        space = 1;
```

```
        for (j = 1; j<= row - 1; j++)
        {
            for (i = 1; i<= space; i++)
            {
                System.out.print(" ");
            }
            space++;
            for (i = 1; i<= 2 * (row - j) - 1; i++)
            {
                System.out.print("*");
            }
            System.out.println("");
        }
    }
}
```

Output-



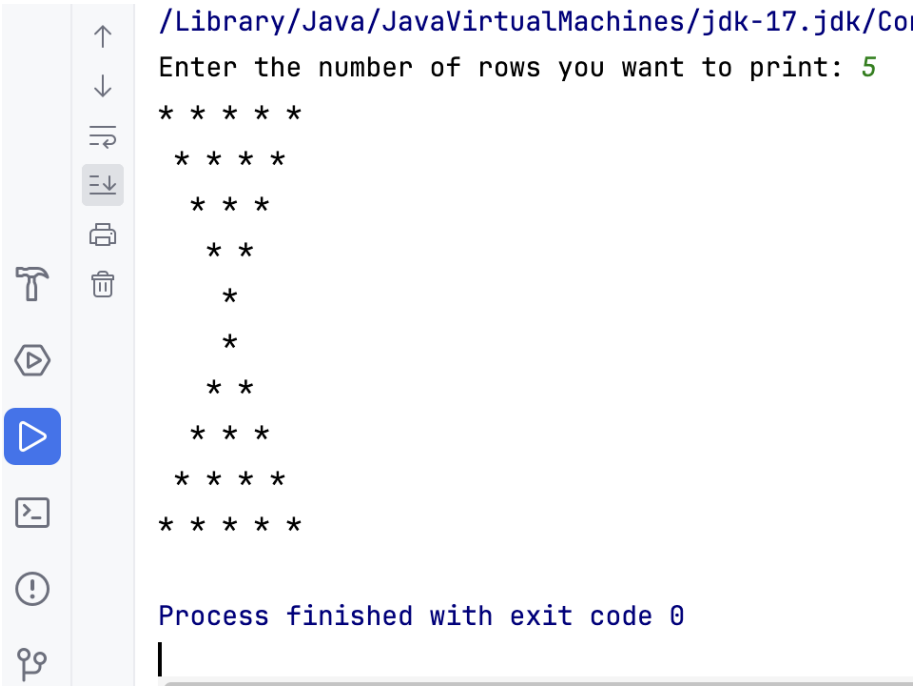
iii) Sandglass Star Pattern

Code-

```
import java.util.Scanner;
public class EXPERIMENT_4_C
{
    public static void main(String[] args)
    {
        int i, j, k, n;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of rows you want
to print: ");
```

```
        n = sc.nextInt();
        for (i= 0; i<= n-1 ; i++)
        {
            for (j=0; j<i; j++)
            {
                System.out.print(" ");
            }
            for (k=i; k<=n-1; k++)
            {
                System.out.print("*" + " ");
            }
            System.out.println("");
        }
        for (i= n-1; i>= 0; i--)
        {
            for (j=0; j<i; j++)
            {
                System.out.print(" ");
            }
            for (k=i; k<=n-1; k++)
            {
                System.out.print("*" + " ");
            }
            System.out.println("");
        }
        sc.close();
    }
}
```

Output-

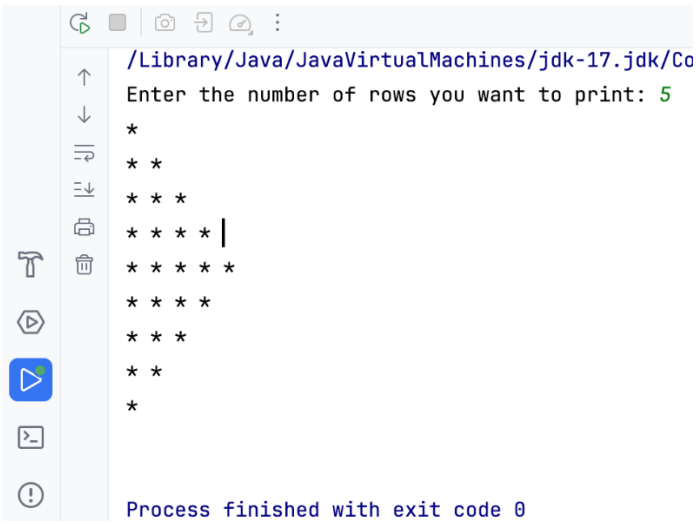


iv) Right Pascal Triangle

Code-

```
import java.util.Scanner;
public class EXPERIMENT_4_D
{
    public static void main(String[] args)
    {
        int i, j, rows;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of rows you want
to print: ");
        rows = sc.nextInt();
        for (i= 0; i<= rows-1; i++)
        {
            for (j=0; j<=i; j++)
            {
                System.out.print("*"+ " ");
            }
            System.out.println("");
        }
        for (i=rows-1; i>=0; i--)
        {
            for(j=0; j <= i-1;j++)
            {
                System.out.print("*"+ " ");
            }
            System.out.println("");
        }
    }
}
```

Output-



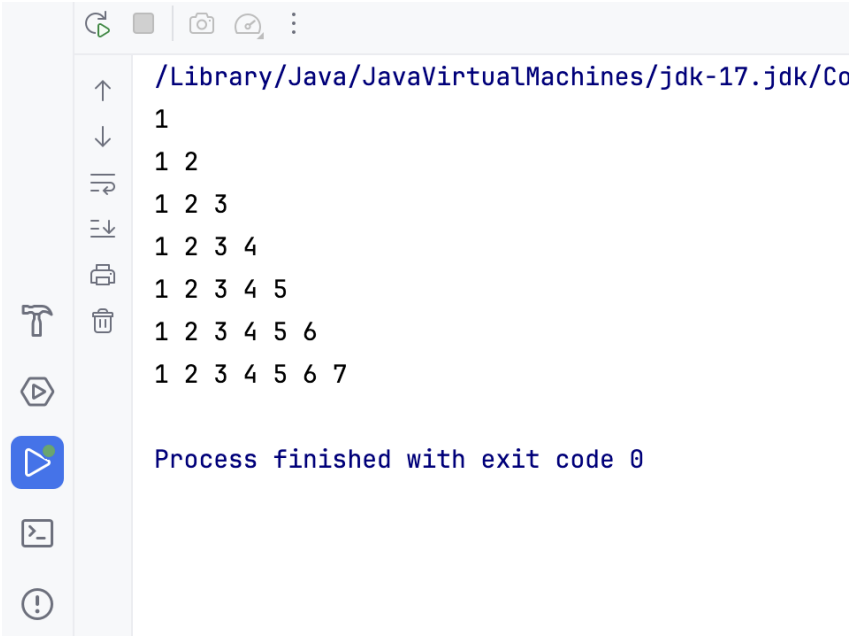
(B) Number Pattern

i) Incremental Number Pattern

Code-

```
public class EXP_4_ii_a
{
    public static void main(String args[])
    {
        int i, j,number, n=7;
        //loop for rows
        for(i=0; i<n; i++)
        {
            number=1;
            //loop for columns
            for(j=0; j<=i; j++)
            {
                //prints num
                System.out.print(number+ " ");
                //incrementing the value of number
                number++;
            }
            //throws the cursor at the next line after
printing each row
            System.out.println();
        }
    }
}
```

Output-

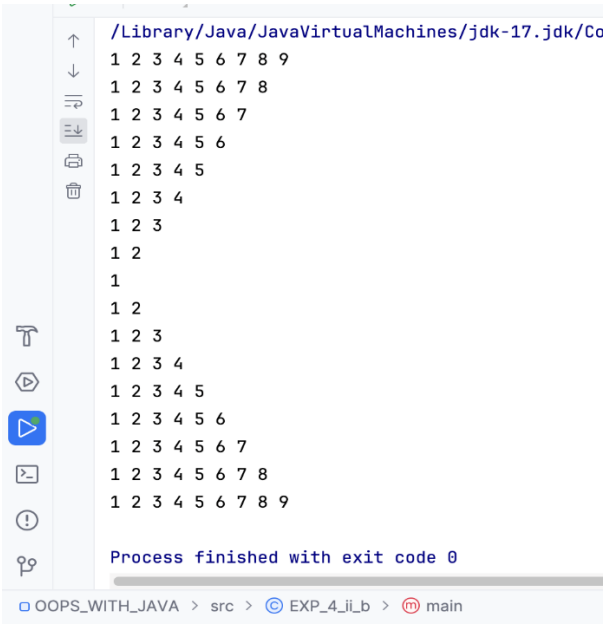


ii) K- Shape Number Pattern

Code-

```
public class EXP_4_ii_b
{
    public static void main(String[] args)
    {
        int i, j, rows=9;
        //Prints upper half  pattern
        for (i = rows; i >= 1; i--)
        {
            for (j = 1; j <= i; j++)
            {
                System.out.print(j+" ");
            }
            System.out.println();
        }
        //Prints lower half  pattern
        for (i = 2; i <= rows; i++)
        {
            for (j = 1; j <= i; j++)
            {
                System.out.print(j+" ");
            }
            System.out.println();
        }
    }
}
```

Output-

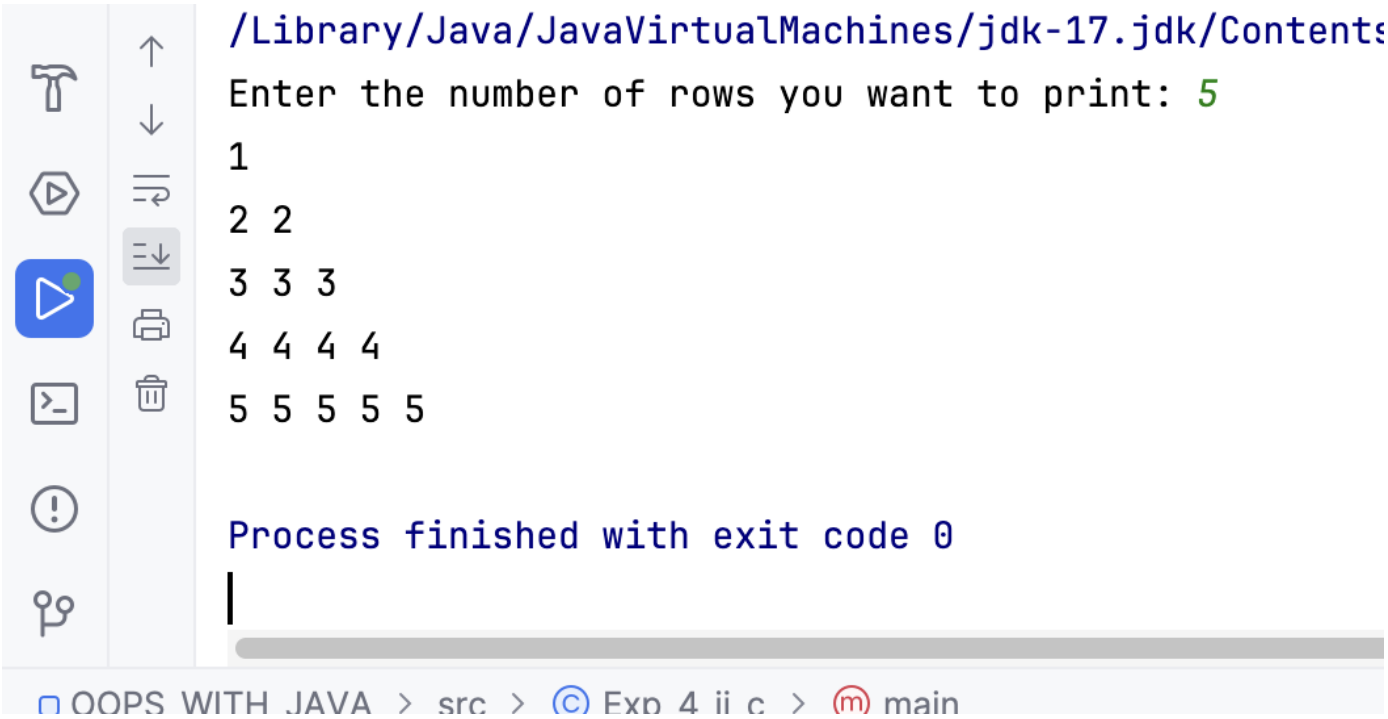


iii) Triangular Number Pattern

Code-

```
import java.util.*;
public class Exp_4_ii_c
{
    public static void main(String[] args)
    {
        int i, j, rows;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of rows you want
to print: ");
        rows = sc.nextInt();
        for (i = 1; i <= rows; i++)
        {
            for (j = 1; j <= i; j++)
            {
                System.out.print(i+" ");
            }
            System.out.println();
        }
    }
}
```

Output-

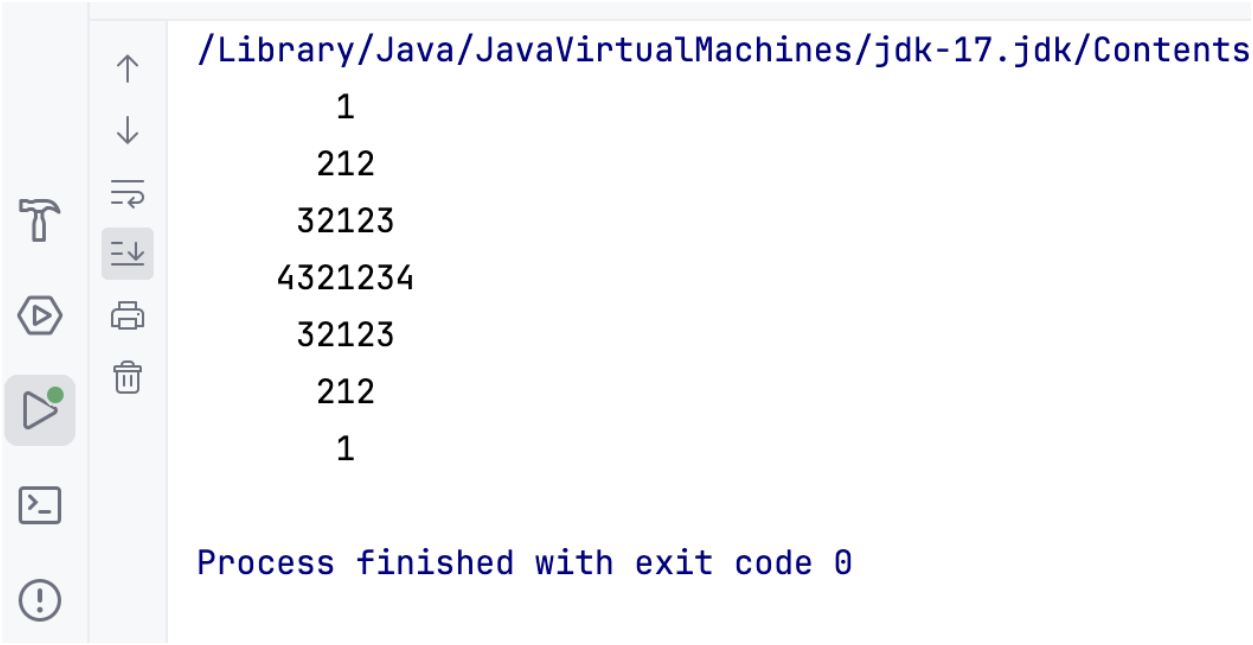


iv) Pyramidal Number Pattern

Code-

```
public class Exp_4_ii_d
{
    public static void main(String[] args)
    {
        for (int i = 1; i <= 4; i++)
        {
            int n = 8;
            for (int j = 1; j<= n - i; j++)
            {
                System.out.print(" ");
            }
            for (int k = i; k >= 1; k--)
            {
                System.out.print(k);
            }
            for (int l = 2; l <= i; l++)
            {
                System.out.print(l);
            }
            System.out.println();
        }
        for (int i = 3; i >= 1; i--)
        {
            int n = 7;
            for (int j = 0; j<= n - i; j++)
            {
                System.out.print(" ");
            }
            for (int k = i; k >= 1; k--)
            {
                System.out.print(k);
            }
            for (int l = 2; l <= i; l++)
            {
                System.out.print(l);
            }
            System.out.println();
        }
    }
}
```

Output-



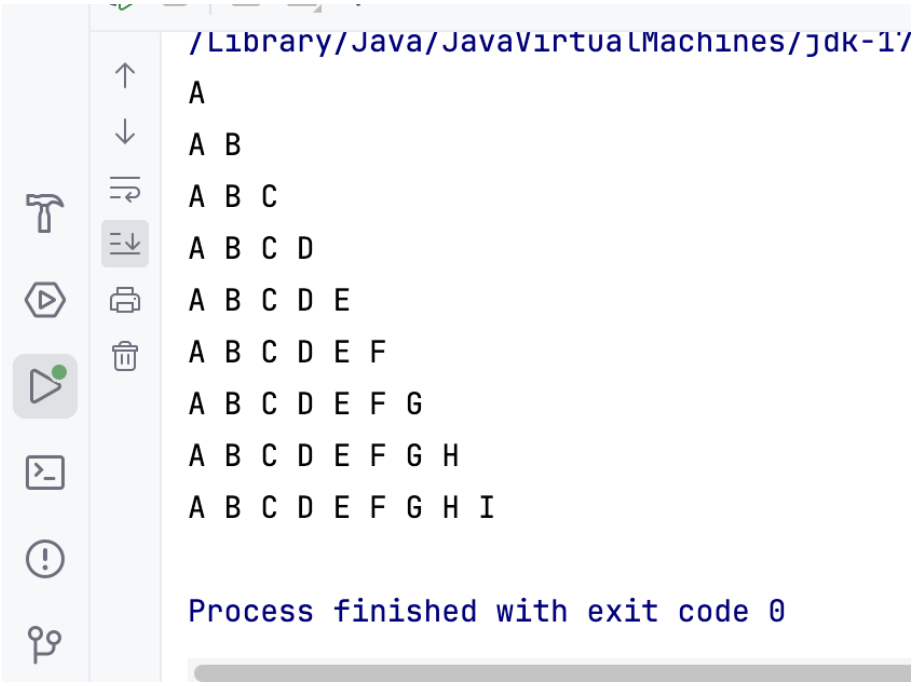
(C) Character Pattern

i) Right Triangle Alphabetic Pattern

Code-

```
public class Exp_4_iii_a
{
    public static void main(String[] args)
    {
        int alphabet = 65; //ASCII value of capital A is 65
        //inner loop for rows
        for (int i = 0; i <= 8; i++)
        {
            //outer loop for columns
            for (int j = 0; j <= i; j++)
            {
                //adds the value of j in the ASCII value of A
and prints the corresponding alphabet
                System.out.print((char) (alphabet + j) + "
");
            }
            System.out.println();
        }
    }
}
```

Output-

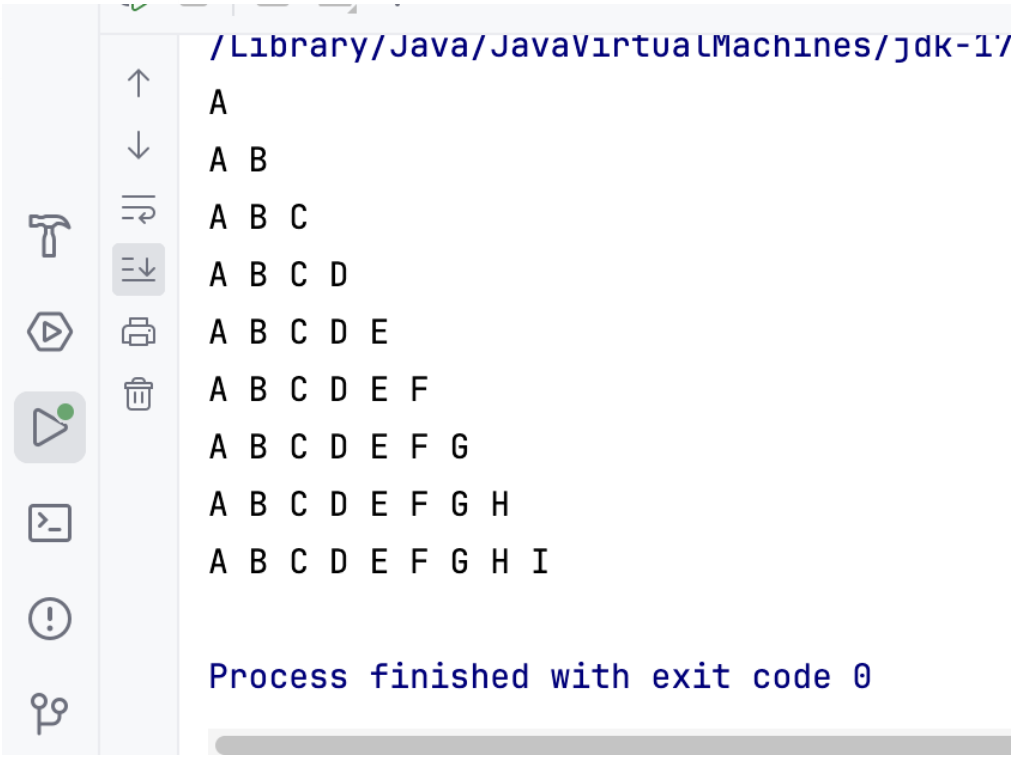


ii) Repeating Alphabet Pattern

Code-

```
public class Exp_4_iii_b
{
    public static void main(String[] args)
    {
        int letter = 65; //ASCII value of capital A is 65
        //inner loop for rwos
        for (int i = 0; i<= 9; i++)
        {
            //outer loop for columns
            for (int j = 0; j <= i; j++)
            {
                //prints the character
                System.out.print((char) letter + " ");
            }
            letter++;
            System.out.println();
        }
    }
}
```

Output-



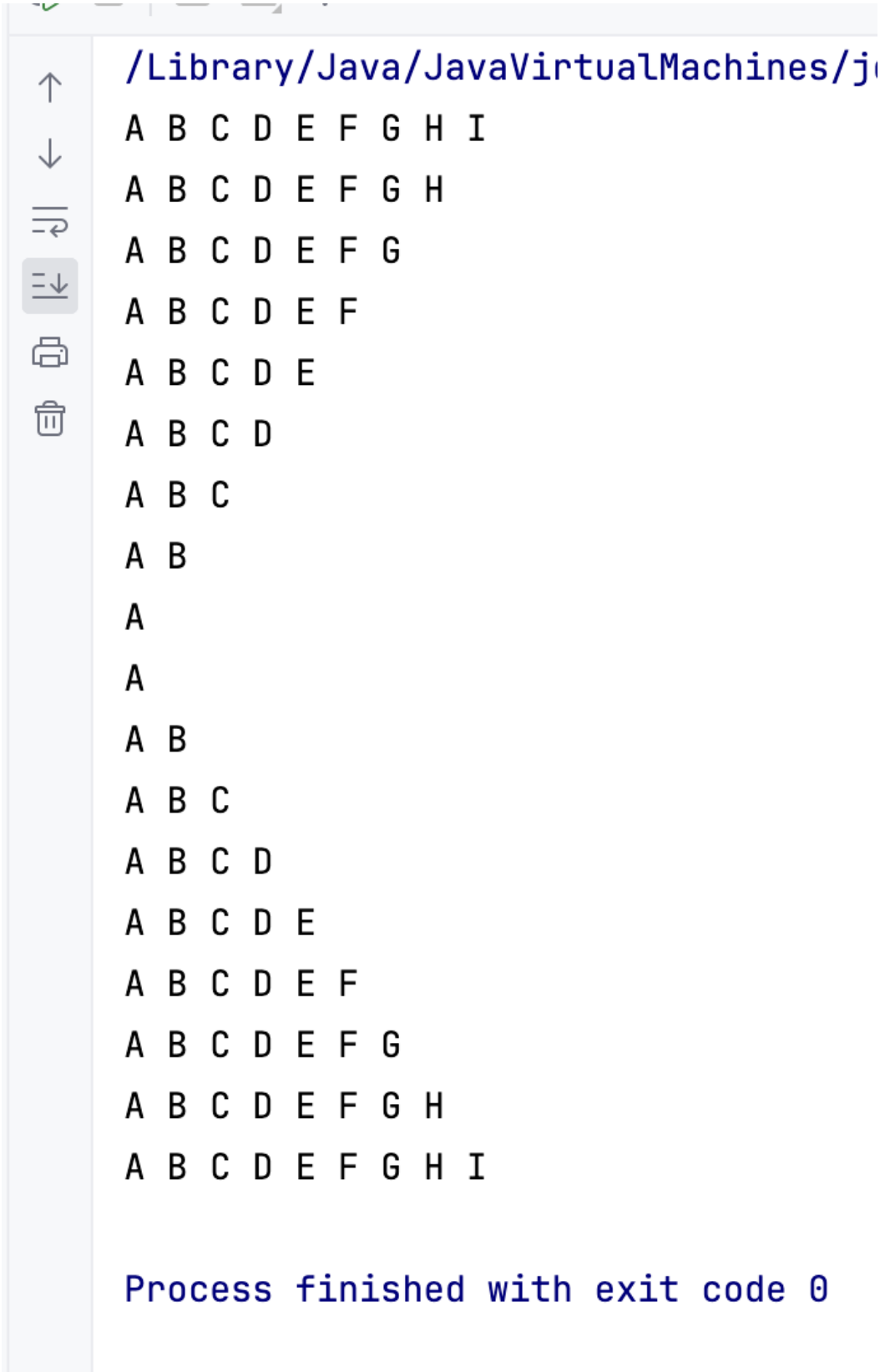
iii) K – Shape Alphabet Pattern

Code–

```
public class Exp_4_iii_c
{
    public static void main(String[] args)
    {
        for (int i = 8; i >= 0; i--)
        {
            int alphabet = 65;
            for (int j = 0; j <= i; j++)
            {
                System.out.print((char) (alphabet + j) + "
");
            }
            System.out.println();
        }
        for (int i = 0; i<= 8; i++)
        {
            int alphabet = 65;
            for (int j = 0; j <= i; j++)
            {
                System.out.print((char) (alphabet + j) + "
");
            }
            System.out.println();
```

```
        }
    }
}
```

Output-

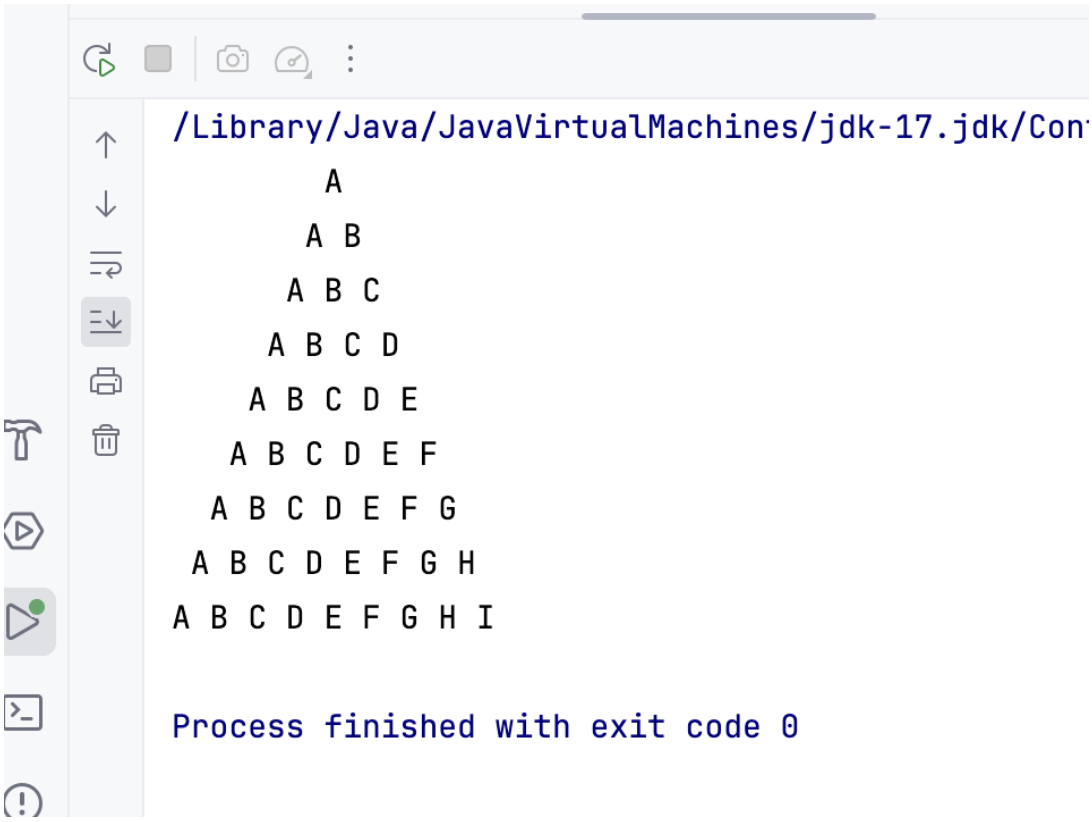

/Library/Java/JavaVirtualMachines/j
A B C D E F G H I
A B C D E F G H
A B C D E F G
A B C D E F
A B C D E
A B C D
A B C
A B
A
A
A B
A B C
A B C D
A B C D E
A B C D E F
A B C D E F G
A B C D E F G H
A B C D E F G H I
Process finished with exit code 0

iv) Triangle Character Pattern

Code-

```
public class Exp_4_iii_D
{
    public static void main(String[] args)
    {
        for (int i = 0; i <= 8; i++)
        {
            int alphabet = 65;
            for (int j = 8; j > i; j--)
            {
                System.out.print(" ");
            }
            for (int k = 0; k <= i; k++)
            {
                System.out.print((char) (alphabet + k) + "
");
            }
            System.out.println();
        }
    }
}
```

Output-



Name :Shubham Dev Enrolment No : 01919011722 Batch : IIOT-B1

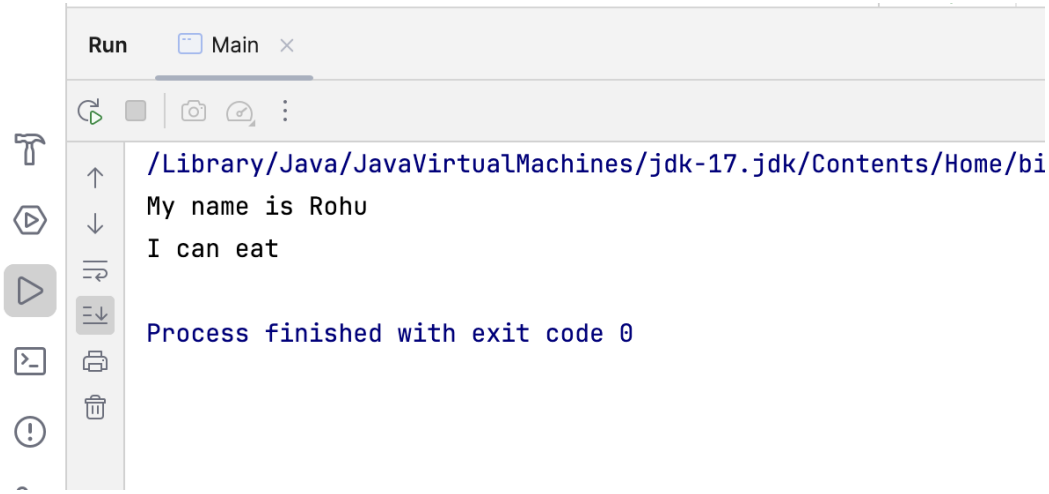
Lab – 5

Aim– (A) Write a program for Java Inheritance

Code –

```
class Animal {
    // field and method of the parent class
    String name;
    public void eat() {
        System.out.println("I can eat");
    }
}
// inherit from Animal
class Dog extends Animal {
    // new method in subclass
    public void display() {
        System.out.println("My name is " + name);
    }
}
class Main {
    public static void main(String[] args) {
        // create an object of the subclass
        Dog labrador = new Dog();
        // access field of superclass
        labrador.name = "Rohu";
        labrador.display();
        // call method of superclass
        // using object of subclass
        labrador.eat();
    }
}
```

Output–

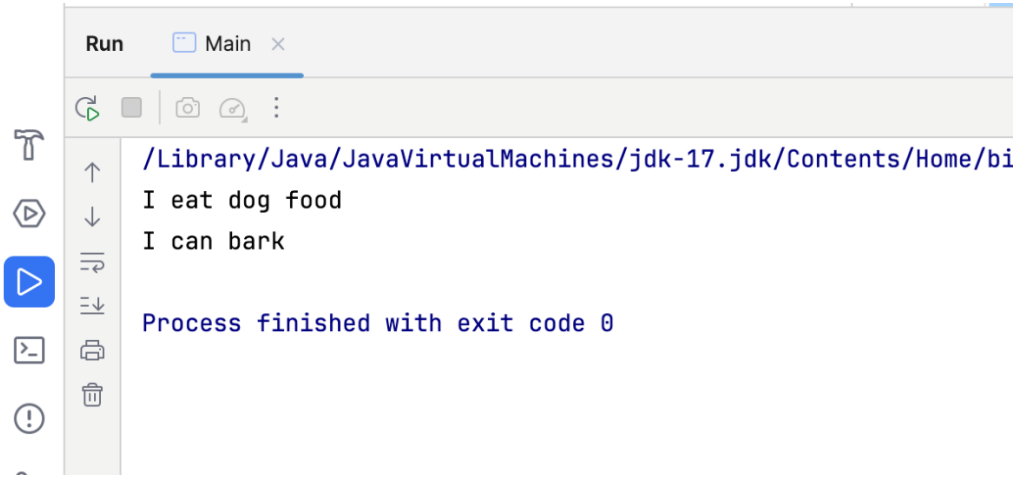


Aim– (B) Write a code for method overriding in Java Inheritance.

Code –

```
class Animal {
    // method in the superclass
    public void eat() {
        System.out.println("I can eat");
    }
}
// Dog inherits Animal
class Dog extends Animal {
    // overriding the eat() method
    @Override
    public void eat() {
        System.out.println("I eat dog food");
    }
    // new method in subclass
    public void bark() {
        System.out.println("I can bark");
    }
}
class Main {
    public static void main(String[] args) {
        // create an object of the subclass
        Dog labrador = new Dog();
        // call the eat() method
        labrador.eat();
        labrador.bark();
    }
}
```

Output –



Name :Shubham Dev Enrolment No : 01919011722 Batch : IIOT-B1

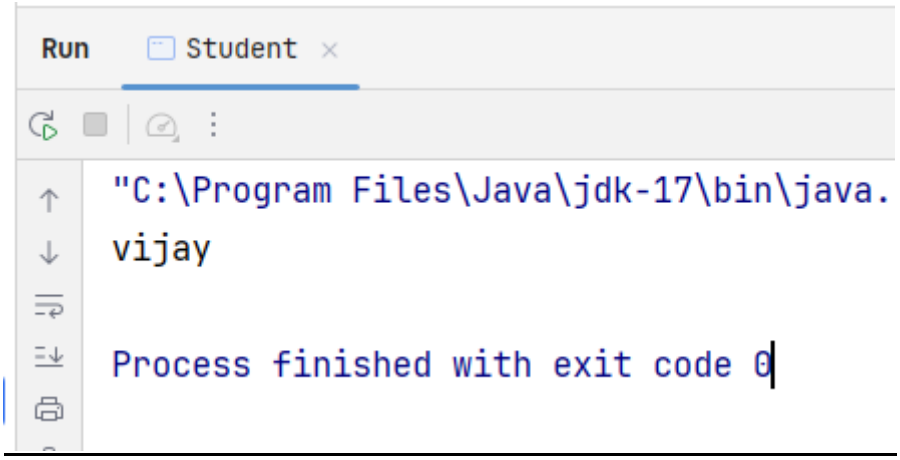
Lab – 6

Aim– (A) WAP to perform the OOPS Encapsulation for read & write mode.

Code –

```
//A Java class which is a fully encapsulated class.
//It has a private data member and getter and setter methods.
public class Student {
    // private data member
    private String name;
    // getter method for name
    public String getName() {
        return name;
    }
    // setter method for name
    public void setName(String name) {
        this.name = name;
    }
    public static void main(String[] args) {
// creating instance of the encapsulated class
        Student s = new Student();
// setting value in the name member
        s.setName("vijay");
// getting value of the name member
        System.out.println(s.getName());
    }
}
```

Output –



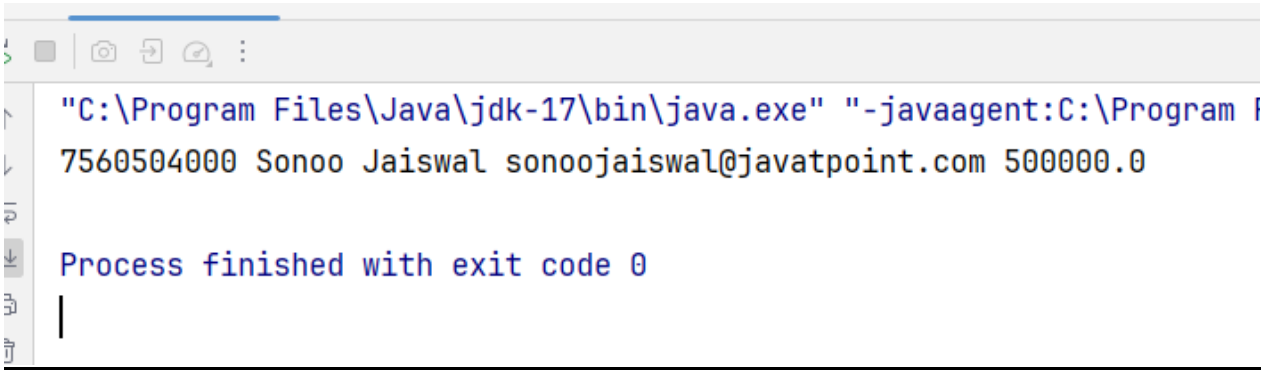
Aim- (B) WAP to perform the OOPS Encapsulation for read & write mode

Code –

```
class Account {
    // private data members
    private long acc_no;
    private String name, email;
    private float amount;
    // public getter and setter methods
    public long getAcc_no() {
        return acc_no;
    }
    public void setAcc_no(long acc_no) {
        this.acc_no = acc_no;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public float getAmount() {
        return amount;
    }public void setAmount(float amount) {
        this.amount = amount;
    }
    public static void main(String[] args) {
// creating instance of Account class
        Account acc = new Account();
// setting values through setter methods
        acc.setAcc_no(7560504000L);
        acc.setName("Sonoo Jaiswal");
        acc.setEmail("sonoojaiswal@javatpoint.com");
        acc.setAmount(500000f);
// getting values through getter methods
        System.out.println(acc.getAcc_no() + " " +
acc.getName() + " " +
                acc.getEmail() + " " + acc.getAmount());
```

```
    }
}
```

Output –


"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program
7560504000 Sonoo Jaiswal sonoojaiswal@javatpoint.com 500000.0
Process finished with exit code 0

Lab – 7

Aim– (A) Write a program to perform polymorphism using compile time method.

Code –

```
// Java Program for Method overloading by using
Different Types of Arguments
// Class 1
// Helper class
class Helper {
    // Method with 2 integer parameters
    static int Multiply(int a, int b)
    {
// Returns product of integer numbers
        return a * b;
    }
    // Method 2
// With same name but with 2 double parameters
    static double Multiply(double a, double b)
    {
// Returns product of double numbers
        return a * b;
    }
}
// Class 2
// Main class
class GFG {
    // Main driver method
    public static void main(String[] args)
    {
// Calling method by passing
// input as in arguments
        System.out.println(Helper.Multiply(2, 4));
        System.out.println(Helper.Multiply(5.5, 6.3));
    }
}
```

Output –

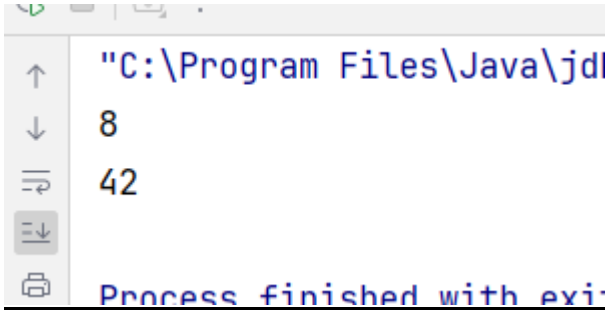
```
"C:\Program Files\Java\jdk-17
8
34.65
```

Aim– (B) Write a program to perform polymorphism using compile time method.

Code –

```
// Java program for Method Overloading by Using
Different Numbers of Arguments
// Class 1
// Helper class
class Helper {
    // Method 1
// Multiplication of 2 numbers
    static int Multiply(int a, int b)
    {
// Return product
        return a * b;
    }
    // Method 2
// // Multiplication of 3 numbers
    static int Multiply(int a, int b, int c)
    {
// Return product
        return a * b * c;
    }
}
// Class 2
// Main class
class GFG {
    // Main driver method
    public static void main(String[] args){
// Calling method by passing
// input as in arguments
        System.out.println(Helper.Multiply(2, 4));
        System.out.println(Helper.Multiply(2, 7, 3));
    }
}
```

Output –



Lab – 8

Aim– (A) Write a program to perform polymorphism using run time method.

Code –

```
class Car {
    void run() {
        System.out.println("running");
    }
}
class Innova extends Car {
    void run() { // Removed semicolon here
        System.out.println("running fast at 120km");
    }
    public static void main(String args[]) {
        Car c = new Innova(); // Changed innova to Innova
(Java is casesensitive)
        c.run();
    }
}
```

Output –

```
"C:\Program Files\Java\jdk-17\bin\java
running fast at 120km

Process finished with exit code 0
```

Aim– (B) Write a program to perform polymorphism using run time method

Code –

```
class Animal {
    void eat() {
        System.out.println("Animals Eat");
    }
}
class Herbivores extends Animal {
    void eat() {
        System.out.println("Herbivores Eat Plants");
    }
}
class Omnivores extends Animal {
    void eat() {
        System.out.println("Omnivores Eat Plants and meat");
}
}
class Carnivores extends Animal {
    void eat() {
        System.out.println("Carnivores Eat meat");}
}
class Main {
    public static void main(String args[]) {
        Animal A = new Animal();
        Animal h = new Herbivores(); //upcasting
        Animal o = new Omnivores(); //upcasting
        Animal c = new Carnivores(); //upcasting
        A.eat();
        h.eat();
        o.eat();
        c.eat();
    }
}
```

Output –