Programming in Java LAB

PAPER CODE	**:**	***CIC-258***
Faculty Name	:	Mr. Anupam Kumar
Name	:	Amit Singhal
Enrollment No.	:	11614802722
Branch	:	Computer Science & Engg.
Semester | Group	:	4C6

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

PSP Area, Plot No. 1, Sector-22, Rohini, Delhi-110086

LAB Assessment Sheet

S.NO.	Experiment Name	M	A	R	K	S	Total Marks	Date of Perf.	Date of Check.	Signature
		R1	R2	R3	R4	R5				
	LAB_01									
1.	Welcome_Message							01-02-24	08-02-24	
2.	ASCII Code of character							01-02-24	08-02-24	
3.	Sum of 2 Integers							01-02-24	08-02-24	
4.	Swap (using Bitwise)							01-02-24	08-02-24	
	LAB_02									
5.	Character_Order							08-02-24	15-02-24	
6.	Colour_Code							08-02-24	15-02-24	
7.	Even_Numbers							08-02-24	15-02-24	
8.	Floyds_Format							08-02-24	15-02-24	
9.	Palindrome_Check							08-02-24	15-02-24	
	LAB_03									
10.	ASCII to Char							15-02-24	22-02-24	
11.	Reverse 2D Array							15-02-24	22-02-24	
12.	Stack using LL							15-02-24	22-02-24	
13.	Queue using LL							15-02-24	22-02-24	
14.	Tokenizer							15-02-24	22-02-24	
15.	Area_Calculator							15-02-24	22-02-24	
	LAB_04									
16.	Box_Volume							22-02-24	29-02-24	
17.	'This' use							22-02-24	29-02-24	
18.	Instance_Counter							22-02-24	29-02-24	
19.	Cube_Calculator							22-02-24	29-02-24	
	LAB_05									
20.	Method_Overriding							22-02-24	29-02-24	
21.	Simple_Inheritance							22-02-24	29-02-24	
22.	MultiLevel_Inheritance							22-02-24	29-02-24	
23.	'super' Keyword							22-02-24	29-02-24	

S.No.	Experiment Name	M	A	R	K	S	Total Marks	Date of Perf.	Date of Check.	Signature
		R1	R2	R3	R4	R5				
24.	Dynamic Polymorphism & Interface Overriding							22-02-24	29-02-24	
	LAB_06									
25.	Abstract Class							29-02-24	07-03-24	
26.	Interface							29-02-24	07-03-24	
27.	Packages							29-02-24	07-03-24	
	LAB_07									
28.	Exception Handling							07-03-24	14-03-24	
29.	Custom Exception							07-03-24	14-03-24	
30.	Applet (Hello World)							07-03-24	14-03-24	
31.	Applet (Analog Clock)							07-03-24	14-03-24	
	LAB_08									
32.	Multi-Threading							14-03-24	21-03-24	
33.	Threading #1							14-03-24	21-03-24	
34.	Threading #2							14-03-24	21-03-24	
35.	File Handling							14-03-24	21-03-24	
	LAB_09									
36.	Slang_Censorer							18-04-24	25-04-24	

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

Computer Science & Engineering Department

VISION

"To be centre of excellence in education, research and technology transfer in the field of computer engineering and promote entrepreneurship and ethical values."

MISSION

To foster an open, multidisciplinary and highly collaborative research environment for producing world-class engineers capable of providing innovative solutions to real-life problems and fulfil societal need.

Lab Exercise - 1

Program - 1

```
//Program to accept a String as a
command-line argument and print a Welcome
message:

package LAB_01;

public class prg_01_welcome {
    public static void main(String[]
args) {
        if (args.length > 0) {
            String name = args[0];
            System.out.println("Welcome "
+ name);
        } else {
            System.out.println("Please
provide your name as a command-line
argument.");
        }
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java> cd LAB_01
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_01> javac prg_01_Welcome_Message.java
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_01> java .\prg_01_Welcome_Message.java AmitSinghal
Welcome AmitSinghal
```

Program - 2

```java
//Program to find ASCII code of a character:

package LAB_01;

import java.util.Scanner;

public class prg_02_ascii {
    public static void main(String[] args) {
        Scanner scanner = new
Scanner(System.in);

        System.out.print("Enter a character:
");
        char inputChar =
scanner.next().charAt(0);

        int asciiValue = (int) inputChar;
        System.out.println("ASCII code of " +
inputChar + " is: " + asciiValue);

        scanner.close();
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_01> java .\prg_02_ASCII_code.java
Enter a character: R
ASCII code of R is: 82
```

Program - 3

```
//Program to accept two integers as inputs and
print their sum:

package LAB_01;

import java.util.Scanner;

public class prg_03_sum {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the first integer:
");
        int num1 = scanner.nextInt ();

        System.out.print("Enter the second
integer: ");
        int num2 = scanner.nextInt();

        int sum = num1 + num2;
        System.out.println("Sum of " + num1 + "
and " + num2 + " is: " + sum);

        scanner.close();
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_01> java .\prg_03_sum.java
Enter the first integer: 116
Enter the second integer: 712
Sum of 116 and 712 is: 828
```

Program - 4

```
//Swapping two numbers using bitwise
operator:

package LAB_01;

public class prg_04_swap {
    public static void main(String[] args) {

        int a = 5;
        int b = 10;

        System.out.println("Before swapping:
a = " + a + ", b = " + b);

        // Using bitwise XOR to swap values
without using a temporary variable
        a = a ^ b;
        b = a ^ b;
        a = a ^ b;

        System.out.println("After swapping: a
= " + a + ", b = " + b);
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_01> java .\prg_04_swap.java
Before swapping: a = 5, b = 10
After swapping: a = 10, b = 5
```

Lab Exercise - 2

Program - 5

```
//Initialize two-character variables in a
program and display the characters in
alphabetical order.

package LAB_02;

public class prg_05_CharacterOrder {
    public static void main(String[] args) {
        char char1 = 'b';
        char char2 = 'a';

        System.out.println("Characters in
alphabetical order:");
        if (char1 < char2) {
            System.out.println(char1 + " " +
char2);
        } else {
            System.out.println(char2 + " " +
char1);
        }
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_02> java .\prg_01_CharacterOrder.java
Characters in alphabetical order:
a b
```

Program - 6

```java
//Write a program to receive a colour
code from the user (an Alphabet).

package LAB_02;

import java.util.Scanner;

public class prg_06_ColourCode {
    public static void main(String[]
args) {
        Scanner scanner = new
Scanner(System.in);

        System.out.print("Enter a colour
code (Alphabet): ");
        char colorCode =
scanner.next().charAt(0);

        System.out.println("Entered
colour code: " + colorCode);

        scanner.close();
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_02> java .\prg_02_ColourCode.java
Enter a colour code (Alphabet): p
Entered colour code: p
```

Program - 7

```
//Write a program to print even
numbers between 23 and 57.

package LAB_02;

public class prg_07_EvenNumbers {
    public static void main(String[]
args) {
        System.out.println("Even
numbers between 23 and 57:");
        for (int i = 23; i <= 57;
i++) {
            if (i % 2 == 0) {
                System.out.print(i);
                System.out.print("
");
            } else
                continue;
        }
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_02> java .\prg_03_EvenNumbers.java
Even numbers between 23 and 57:
24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56
```

Program - 8

```
//Write a program to print in Floyd's
format (using for and while loop).

package LAB_02;

public class prg_08_FloydsFormat {
    public static void main(String[]
args) {
        int n = 5; // Change this value
for a different number of rows
        int num = 1;

        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= i; j++)
{
                System.out.print(num + "
");
                num++;
            }
            System.out.println();
        }
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_02> java .\prg_04_FloydsFormat.java
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

Program - 9

```java
//Write a Java program to find if the given number is palindrome
or not.

package LAB_02;
import java.util.Scanner;

public class prg_09_PalindromeCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        if (isPalindrome(number))
            System.out.println(number + " is a palindrome.");
        else
            System.out.println(number + " is not a palindrome.");
        scanner.close();
    }
    private static boolean isPalindrome(int num) {
        int originalNum = num;
        int reversedNum = 0;
        while (num > 0) {
            int digit = num % 10;
            reversedNum = reversedNum * 10 + digit;
            num /= 10;
        }
        return originalNum == reversedNum;
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_02> java .\prg_05_PalindromeCheck.java
Enter a number: 10101
10101 is a palindrome.
```

Lab Exercise - 3

Program - 10

```
//Initialize an integer array with ASCII values and print the corresponding character values in a single row.

package LAB_03;

public class prg_10_AsciiToChar {
    public static void main(String[] args) {
        int[] asciiArray = { 65, 66, 67, 97, 98, 99 }; // Example ASCII values

        System.out.print("Corresponding characters: ");
        for (int asciiValue : asciiArray) {
            System.out.print((char) asciiValue + " ");
        }
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_03> java .\prg_01_AsciiToChar.java
Corresponding characters: A B C a b c
```

Program – 11

```
//Write a program to reverse the elements of
a given 2*2 array. Four integer numbers need
to be passed as Command-Line arguments.

package LAB_03;

public class prg_11_Reverse2DArray {

    public static void main(String[] args) {
        if (args.length != 4) {
            System.out.println("Please
provide exactly 4 integers as command-line
arguments.");
            return;
        }

        int[][] matrix = { {
Integer.parseInt(args[0]),
Integer.parseInt(args[1]) },
                { Integer.parseInt(args[2]),
Integer.parseInt(args[3]) } };

        System.out.println("Original 2*2
Array:");
        print2DArray(matrix);

        System.out.println("Reversed 2*2
Array:");
        reverse2DArray(matrix);
        print2DArray(matrix);
```

```java
    }

    private static void reverse2DArray(int[][] array) {
        int temp = array[0][0];
        array[0][0] = array[1][1];
        array[1][1] = temp;

        temp = array[0][1];
        array[0][1] = array[1][0];
        array[1][0] = temp;
    }

    private static void print2DArray(int[][] array) {
        for (int[] row : array) {
            for (int element : row) {
                System.out.print(element + " ");
            }
            System.out.println();
        }
    }

}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_03> java .\prg_02_Reverse2DArray.java 10 20 30 40
Original 2*2 Array:
10 20
30 40
Reversed 2*2 Array:
40 30
20 10
```

Program - 12

```
package LAB_03;

class Node {
    int data;
    Node next;

    public Node(int d) {
        data = d;
        next = null;
    }
}

public class prg_12_Stack_LL {
    static class Stacks {
        public static Node Top;
        public static int Size;

        boolean IsEmpty() {
            return (Top == null);
        }

        void Push(int data) {
            Node NewNode = new Node(data);
            if (IsEmpty()) {
                Top = NewNode;
                Size++;
                return;
            }
            NewNode.next = Top;
            Top = NewNode;
```

```
            Size++;
        }

        void Pop() {
            if (IsEmpty()) {
                System.out.println("THE STACK
IS EMPTY , DELETION NOT POSSIBLE");
                return;
            }
            Node Temp = Top;
            Top = Top.next;
            Temp.next = null;
            Size--;
        }

        void Peek() {
            if (IsEmpty()) {
                System.out.println("THE STACK
IS EMPTY , PEEK NOT POSSIBLE");
                return;
            }
            System.out.println("THE TOP OF
THE IS ::: " + Top.data);
        }

        void Size_Stack() {
            System.out.println("THE SIZE OF
THE STACK IS ::: " + Size);
        }

        void Display() {
            System.out.print("THE STACK IS
::: (Top) --> ");
```

```
            Node temp = Top;
            while (temp != null) {
                System.out.print(temp.data +
" ");
                temp = temp.next;
            }
            System.out.println();
        }
    }

    public static void main(String args[]) {
        Stacks S1 = new Stacks();
        System.out.println();
        System.out.println();
        S1.Push(1);
        S1.Push(2);
        S1.Push(3);
        S1.Push(4);
        S1.Push(5);
        S1.Display();
        S1.Pop();
        S1.Display();
        S1.Peek();
        S1.Size_Stack();
        System.out.println();
    }
}
```

```
THE STACK IS ::: (Top) --> 5 4 3 2 1
THE STACK IS ::: (Top) --> 4 3 2 1
THE TOP OF THE IS ::: 4
THE SIZE OF THE STACK IS ::: 4
```

Program - 13

```java
package LAB_03;

class Node {
    int data;
    Node next;
    public Node(int d) {
        data = d;
        next = null;
    }
}

public class prg_13_Queue_LL {
    static class Queues {
        public static Node Front;
        public static Node Rear;
        public static int Size;
        boolean IsEmpty() {
            return (Rear == null);
        }
        void Enqueue(int data) {
            Node NewNode = new Node(data);
            if (IsEmpty()) {
                Front = Rear = NewNode;
                Size++;
                return;
            }
            Rear.next = NewNode;
            Rear = NewNode;
            Size++;
        }

        void Dequeue() {
            if (IsEmpty()) {
```

```
                System.out.println("THE STACK IS
EMPTY , DELETION NOT POSSIBLE");
                return;
            }
            Node Temp = Front;
            Front = Front.next;
            Temp.next = null;
            Size--;
        }

        void Peek_First() {
            if (IsEmpty()) {
                System.out.println("THE QUEUE IS
EMPTY , PEEK NOT POSSIBLE");
                return;
            }
            System.out.println("THE FRONT OF THE
QUEUE IS ::: " + Front.data);
        }
        void Peek_Last() {
            if (IsEmpty()) {
                System.out.println("THE QUEUE IS
EMPTY , PEEK NOT POSSIBLE");
                return;
            }
            System.out.println("THE REAR OF THE
QUEUE IS ::: " + Rear.data);
        }
        void Size_Queue() {
            System.out.println("THE SIZE OF THE
QUEUE IS ::: " + Size);
        }
        void Display() {
            System.out.print("THE QUEUE IS :::
(Front) --> ");
            Node temp = Front;
```

```java
            while (temp != null) {
                System.out.print(temp.data + " ");
                temp = temp.next;
            }
            System.out.println("<-- (Rear)");
        }
    }

    public static void main(String args[]) {
        Queues q = new Queues();
        System.out.println();
        System.out.println();
        q.Enqueue(1);
        q.Enqueue(2);
        q.Enqueue(3);
        q.Enqueue(4);
        q.Enqueue(5);
        q.Display();
        q.Dequeue();
        q.Display();
        q.Peek_First();
        q.Peek_Last();
        q.Size_Queue();
        System.out.println();
    }
}
```

```
THE QUEUE IS ::: (Front) --> 1 2 3 4 5 <-- (Rear)
THE QUEUE IS ::: (Front) --> 2 3 4 5 <-- (Rear)
THE FRONT OF THE QUEUE IS ::: 2
THE REAR OF THE QUEUE IS ::: 5
THE SIZE OF THE QUEUE IS ::: 4
```

Program - 14

```java
//Write a Java program to produce the tokens from the given long string.

package LAB_03;

import java.util.StringTokenizer;

public class prg_14_Tokenizer {
    public static void main(String[] args) {
        String longString = "This is a long string with multiple words.";

        StringTokenizer tokenizer = new StringTokenizer(longString);

        System.out.println("Tokens:");
        while (tokenizer.hasMoreTokens()) {
            System.out.println(tokenizer.nextToken());
        }
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_03> java .\prg_04_Tokenizer.java
Tokens:
This
is
a
long
string
with
multiple
words.
```

Program – 15

```
//Using the concept of method
overloading, write methods for
calculating the area of a triangle,
circle, and rectangle.

package LAB_03;

public class prg_15_AreaCalculator
{
    public static void
main(String[] args) {
        // Example usage
        System.out.println("Area of
Triangle: " + calculateArea(5.324,
8.765));
        System.out.println("Area of
Circle: " + calculateArea(3.5));
        System.out.println("Area of
Rectangle: " + calculateArea(4,
6));
    }

    // Area of Triangle
```

```java
    private static double calculateArea(double base, double height) {
        return 0.5 * base * height;
    }

    // Area of Circle
    private static double calculateArea(double radius) {
        return Math.PI * radius * radius;
    }

    // Area of Rectangle
    private static double calculateArea(int length, int width) {
        return length * width;
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_03> java .\prg_05_AreaCalculator.java
Area of Triangle: 23.332430000000002
Area of Circle: 38.48451000647496
Area of Rectangle: 24.0
```

Lab Exercise - 4

Program - 16

```
//Create a class Box with a
parameterized constructor and a
method to calculate the volume:

package LAB_04;

public class prg_16_Box {
    private double width;
    private double height;
    private double depth;

    // Parameterized constructor
    public prg_16_Box(double width,
double height, double depth) {
        this.width = width;
        this.height = height;
        this.depth = depth;
    }
```

```java
    // Method to calculate volume
    public double calculateVolume()
{
        return width * height *
depth;
    }

    public static void
main(String[] args) {
        // Create an object of the
Box class
        prg_16_Box myBox = new
prg_16_Box(3.0, 4.0, 5.0);

        // Test the functionalities
        System.out.println("Volume
of the box: " +
myBox.calculateVolume());
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_04> java .\Box.java
Volume of the box: 60.0
```

Program - 17

```
//Write a program to display the use of this
keyword:

package LAB_04;

public class prg_17_This {
    private int value;

    // Parameterized constructor using this
keyword
    public prg_17_This(int value) {
        this.value = value;
    }

    // Method using this keyword
    public void displayValue() {
        System.out.println("Value: " +
this.value);
    }

    public static void main(String[] args) {
        // Create an object of the class
        prg_17_This obj = new prg_17_This(42);

        // Call the method to display the value
        obj.displayValue();
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_04> java .\This.java
Value: 42
```

Program - 18

```
//Write a program to count the number of instances
created for the class:

package LAB_04;

public class prg_18_InstanceCounter {
    private static int instanceCount = 0;

    // Constructor increments the instance count
    public prg_18_InstanceCounter() {
        instanceCount++;
    }

    // Static method to get the instance count
    public static int getInstanceCount() {
        return instanceCount;
    }

    public static void main(String[] args) {
        // Create instances of the class
        prg_18_InstanceCounter obj1 = new
prg_18_InstanceCounter();
        prg_18_InstanceCounter obj2 = new
prg_18_InstanceCounter();

        // Get and display the instance count
        System.out.println("Number of instances created:
" + prg_18_InstanceCounter.getInstanceCount());
    }
}
```

```
PS C:\Users\Admin\Dc
Number of instances
```

Program - 19

```java
//Java program to get the cube of a given number using a static method:

package LAB_04;

public class prg_19_CubeCalculator {
    // Static method to calculate the cube
    public static double calculateCube(double number) {
        return Math.pow(number, 3);
    }

    public static void main(String[] args) {
        // Test the static method
        double result = prg_19_CubeCalculator.calculateCube(4.0);
        System.out.println("Cube of the given number: " + result);
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java\LAB_04> java .\CubeCalculator.java
Cube of the given number: 64.0
```

Lab Exercise - 5

Program – 20

```java
//Implement Method Overriding:

package LAB_05;

class Animal {
    void sound() {
        System.out.println("Animal makes a sound");
    }
}

class Dog extends Animal {
    void sound() {
        System.out.println("Dog barks");
    }
}

public class prg_20_MethodOverriding {
    public static void main(String[] args) {
        Animal animal = new Dog();
        animal.sound(); // Calls the overridden method in
Dog class
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java>  & 'C:\Users\Admin\AppData\Lo
 '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Admin\AppData\
c7\redhat.java\jdt_ws\Java_e1c454ad\bin' 'LAB_05.prg_20_MethodOverriding'
Dog barks
```

Program – 21

```
//Illustrate Simple Inheritance:

package LAB_05;

class Parent {
    void display() {
        System.out.println("This is the parent
class");
    }
}

class Child extends Parent {
    void show() {
        System.out.println("This is the child
class");
    }
}

public class prg_21_SimpleInheritance {
    public static void main(String[] args) {
        Child childObj = new Child();
        childObj.display(); // Accessing method
from the parent class
        childObj.show(); // Accessing method from
the child class
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java>  & 'C:\Users\Admin\AppData\Loc
 '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Admin\AppData\R
c7\redhat.java\jdt_ws\Java_e1c454ad\bin' 'LAB_05.prg_21_SimpleInheritance'
This is the parent class
This is the child class
```

Program – 22

```
//Illustrate Multilevel Inheritance:

package LAB_05;

class Vehicle {
    void start() {
        System.out.println("Vehicle
started");
    }
}

class Car extends Vehicle {
    void accelerate() {
        System.out.println("Car is
accelerating");
    }
}

class SportsCar extends Car {
    void turboCharge() {
        System.out.println("Sports car
turbocharged");
    }
}
```

```java
public class
prg_22_MultilevelInheritance {
    public static void main(String[]
args) {
        SportsCar sportsCarObj = new
SportsCar();
        sportsCarObj.start(); //
Accessing method from the Vehicle
class
        sportsCarObj.accelerate(); //
Accessing method from the Car class
        sportsCarObj.turboCharge(); //
Accessing method from the SportsCar
class
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java>  & 'C:\Users\Admin\AppData\Local\P
e Adoptium\jdk-17.0.10.7-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptio
p' 'C:\Users\Admin\AppData\Roaming\Code\User\workspaceStorage\bd2e67f6d49f044c
c7\redhat.java\jdt_ws\Java_e1c454ad\bin' 'LAB_05.prg_22_MultilevelInheritance'
Vehicle started
Car is accelerating
Sports car turbocharged
```

Program – 23

```
// Illustrate all Uses of super
Keyword:

package LAB_05;

class Base {
    int x = 10;

    void display() {
        System.out.println("This is
the Base class");
    }
}

class Derived extends Base {
    int x = 20;

    void show() {
        int x = 30;
        System.out.println("Local
variable x: " + x);
```

```java
        System.out.println("Derived class variable x: " + this.x);
        System.out.println("Base class variable x: " + super.x);
        super.display(); // Calls the method from the Base class using super
    }
}

public class prg_23_SuperKeyword {
    public static void main(String[] args) {
        Derived derivedObj = new Derived();
        derivedObj.show();
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java>  & 'C:\Users\Admin\AppData\Local\Programs\Eclipse Ad
bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Admin\AppData\Roaming\Co
67f6d49f044c9ec56649cd963ec7\redhat.java\jdt_ws\Java_e1c454ad\bin' 'LAB_05.prg_23_SuperKeyword'
Local variable x: 30
Derived class variable x: 20
Base class variable x: 10
This is the Base class
```

Program – 24

```
//Show Dynamic Polymorphism and
Interface Overriding:

package LAB_05;

interface Shape {
    void draw();
}

class Circle implements Shape {
    @Override
    public void draw() {
        System.out.println("Drawing
Circle");
    }
}

class Rectangle implements Shape {
    @Override
    public void draw() {
        System.out.println("Drawing
Rectangle");
    }
```

```
}

public class
prg_24_DynamicPolymorphism {
    public static void
main(String[] args) {
        Shape circle = new
Circle();
        Shape rectangle = new
Rectangle();

        circle.draw(); // Calls
Circle's implementation of draw()
        rectangle.draw(); // Calls
Rectangle's implementation of
draw()
    }
}
```

```
PS C:\Users\Admin\Downloads\#LAB WORK\Java>  & 'C:\Users\Admin\AppData\Local
e Adoptium\jdk-17.0.10.7-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExcept
p' 'C:\Users\Admin\AppData\Roaming\Code\User\workspaceStorage\bd2e67f6d49f04
c7\redhat.java\jdt_ws\Java_e1c454ad\bin' 'LAB_05.prg_24_DynamicPolymorphism'
Drawing Circle
Drawing Rectangle
```

Lab Exercise - 6

Program – 25

```
//WAP to create an abstract class "Shape" where
"Rectangle" & "Triangle" inherit the "Shape" class.

package LAB_06;

abstract class Shape {
    int Base, Height;

    public abstract void Show();

    public abstract void Area();
}

class Rectangle extends Shape {
    public Rectangle(int b, int h) {
        Base = b;
        Height = h;
    }

    @Override
    public void Show() {
        System.out.println("The Rectangle Has Base = " +
Base + " & Height = " + Height);
    }

    public void Area() {
        System.out.println("The Area Of Rectangle is -> "
+ (Base * Height));
    }
}

class Triangle extends Shape {
    public Triangle(int b, int h) {
        Base = b;
```

```
        Height = h;
    }

    @Override
    public void Show() {
        System.out.println("The Triangle Has Base = " +
Base + " & Height = " + Height);
    }

    public void Area() {
        System.out.println("The Area Of Triangle is -> "
+ (0.5 * Base * Height));
    }
}

public class prg_25_AbstractClass {
    public static void main(String args[]) {
        System.out.println();
        Shape s = new Rectangle(10, 20);
        s.Show();
        s.Area();
        System.out.println();
        s = new Triangle(20, 20);
        s.Show();
        s.Area();
        System.out.println();
    }
}
```

```
The Rectangle Has Base = 10 & Height = 20
The Area Of Rectangle is -> 200

The Triangle Has Base = 20 & Height = 20
The Area Of Triangle is -> 200.0
```

Program – 26

```java
//WAP that creates an Interface & implements it

package LAB_06;

// Define the interface
interface MyInterface {
    // Abstract method declaration
    void myMethod();
}

// Implement the interface in a class
class MyClass implements MyInterface {
    // Implementing the abstract method from the
interface
    public void myMethod() {
        System.out.println("Implementing
myMethod() in MyClass");
    }
}

public class prg_26_Interface {
    public static void main(String[] args) {
        // Create an object of the implementing
class
        MyClass obj = new MyClass();

        // Call the method implemented from the
interface
        obj.myMethod();
    }
}
```

```
Implementing myMethod() in MyClass
```

Program – 27

```
/*
Write an interface "playable" with a method void
"play()", let this Interface be placed in a package
called "music".
Write a class "Veena" which implement the "playable"
interface, let this class be placed in a package called
"music.string".
Write a class "saxophone" which implement the "playable"
interface, let this class be placed in a package called
"music.wind".
Write another class "test" in package "live". Then ->
(i) create an instance of "Veena" and call the "play()"
method
(ii) create an instance of "saxophone" and Call the
"play()" method
(iii) place the above instances in a variable of type
"playable" and then call "play())"
*/
```

playable.java

```java
package LAB_06.prg_27_PackageInterface.live.music;

public interface playable {
    void play();
}
```

veena.java

```java
package
LAB_06.prg_27_PackageInterface.live.music.string;
```

```java
import
LAB_06.prg_27_PackageInterface.live.music.playable
;

public class veena implements playable {
    String name;

    public veena(String n) {
        this.name = n;
    }

    @Override
    public void play() {
        System.out.println(name + "can play veena
well!");
    }

    public static void main(String[] args) {
        playable p = new veena("Amit ");
        p.play();
    }
}
```

saxophone.java

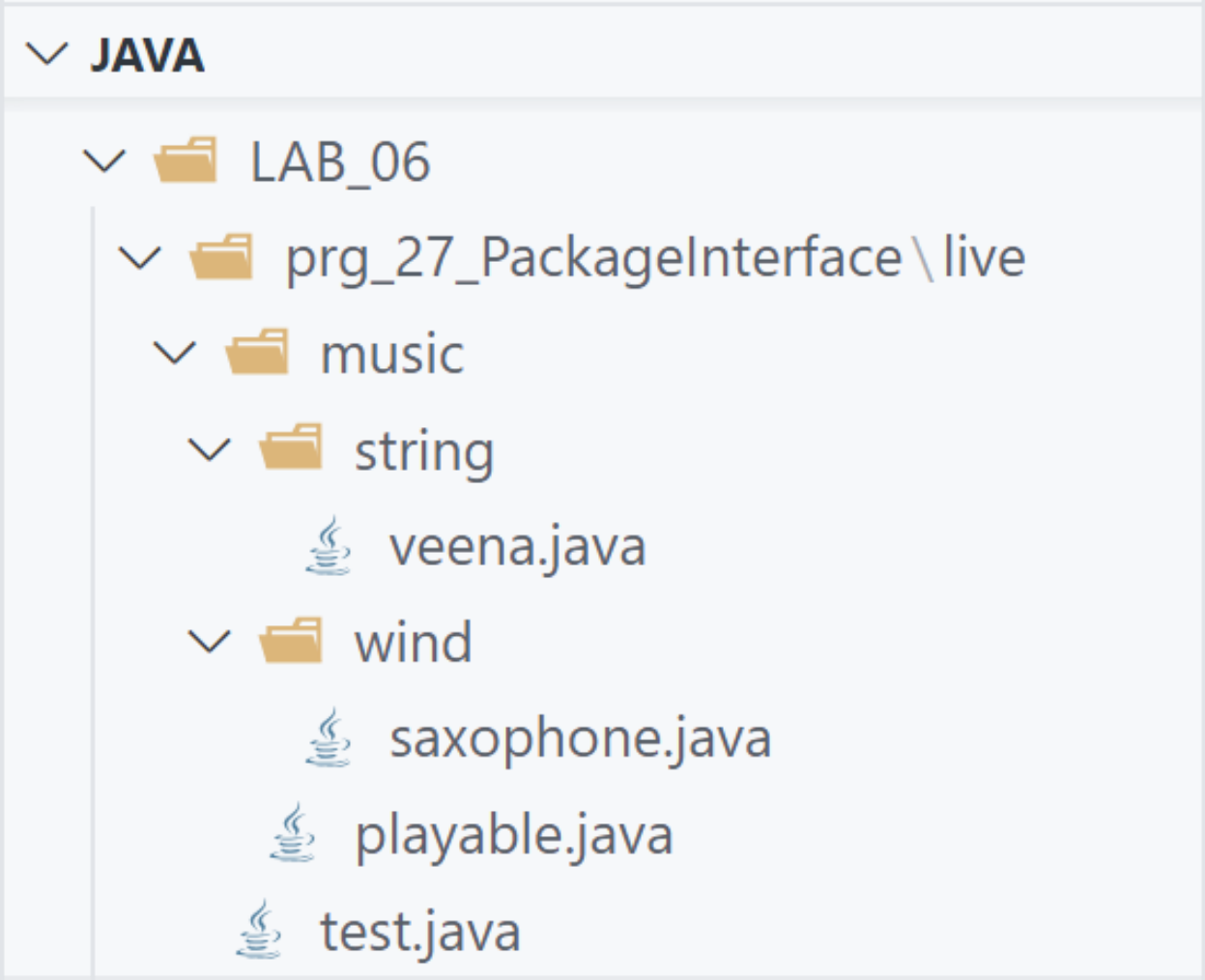
```java
package
LAB_06.prg_27_PackageInterface.live.music.wind;

import
LAB_06.prg_27_PackageInterface.live.music.playable
;

public class saxophone implements playable {
    String name;

    public saxophone(String n) {
```

```
        this.name = n;
    }

    @Override
    public void play() {
        System.out.println(name + "can play
saxophone well!");
    }

    public static void main(String[] args) {
        playable p = new saxophone("Amit ");
        p.play();
    }
}
```

test.java

```
package LAB_06.prg_27_PackageInterface.live;

import
LAB_06.prg_27_PackageInterface.live.music.playable
;
import
LAB_06.prg_27_PackageInterface.live.music.string.v
eena;
import
LAB_06.prg_27_PackageInterface.live.music.wind.sax
ophone;

public class test {
    public static void main(String[] args) {
        System.out.print("Instance of Veena Class
: ");
        veena v = new veena("Amit ");
        v.play();
```

```java
        System.out.print("Instance of Saxophone
Class : ");
        saxophone s = new saxophone("Amit ");
        s.play();
        System.out.println("Instance of Playable
Interface ->");
        playable p = new veena("Amit ");
        p.play();
        p = new saxophone("Amit ");
        p.play();
    }
}
```




```
Instance of Veena Class : Amit can play veena well!
Instance of Saxophone Class : Amit can play saxophone well!
Instance of Playable Interface ->
Amit can play veena well!
Amit can play saxophone well!
```

Lab Exercise - 7

Program – 28

```
//WAP to accept name & age of a person from the user.
Ensure that entered age is between 15 & 60. Display
proper error message & the program must execute
gracefully after displaying the error message in case the
argument pass is not proper.

package LAB_07;

import java.util.*;

class MyException extends Exception {
    public MyException(String message) {
        super(message);
    }
}

public class prg_28_Exception {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println();
        String name;
        int Age;
        try {
            System.out.print("Enter your Name -> ");
            name = sc.nextLine();
            System.out.print("Enter your Age -> ");
            Age = sc.nextInt();
            System.out.println();
            if (Age < 15) {
                throw new MyException(name + " Age is " +
Age + " Which Is Less Than 15 Years");

            } else if (Age > 60) {
```

```
                throw new MyException(name + " Age is " +
Age + " Which Is More Than 60 Years");

            } else {
                System.out.println(name + " Age is ::: "
+ Age);

            }
        } catch (Exception e) {
            System.out.println("Error : " + e);
        } finally {
            System.out.println("Finally Program is
Finished...");
            System.out.println();
            sc.close();
        }
    }
}
```

```
Enter your Name -> Amit
Enter your Age -> 14
Error : LAB_07.MyException: Amit Age is 14 Which Is Less Than 15 Years
Finally Program is Finished...
```

```
Enter your Name -> Shaswat
Enter your Age -> 20

Shaswat Age is ::: 20
Finally Program is Finished...
```

```
Enter your Name -> Dadaji
Enter your Age -> 69

Error : LAB_07.MyException: Dadaji Age is 69 Which Is More Than 60 Years
Finally Program is Finished...
```

Program – 29

```
//WAP to create a customized exception & also make use of all the 5
exception keywords.

package LAB_07;

class MyException extends Exception {
    public MyException(String message) {
        super(message);
    }
}

public class prg_29_CustomException {
    public static void CheckException(String Name, int Age) {
        try {
            Check(Name, Age);
        } catch (MyException e) {
            System.out.println("Error : " + e);
        } finally {
            System.out.println("Finally Program Executed...");
            System.out.println();
        }
    }

    public static void Check(String Name, int Age) throws MyException {
        if (Age < 18) {
            throw new MyException(Name + " Is Not Eligible For Voting");
        } else {
            System.out.println(Name + " Is Eligible For Voting");
        }
    }

    public static void main(String args[]) {
        CheckException("Amit", 21);
        CheckException("Justin", 17);
    }
}
```

```
Amit Is Eligible For Voting
Finally Program Executed...

Error : LAB_07.MyException: Justin Is Not Eligible For Voting
Finally Program Executed...
```

Program – 30

//Write an applet prg that displays "Hello World" with background color "black", text color "blue" and your name in the status window.

.java file

```java
package LAB_07;

import java.applet.*;
import java.awt.*;

public class prg_30_Applet1 extends Applet {
    public void paint(Graphics g) {
        g.setColor(Color.blue);
        g.drawString("Hello World", 50, 50);
    }
}
```

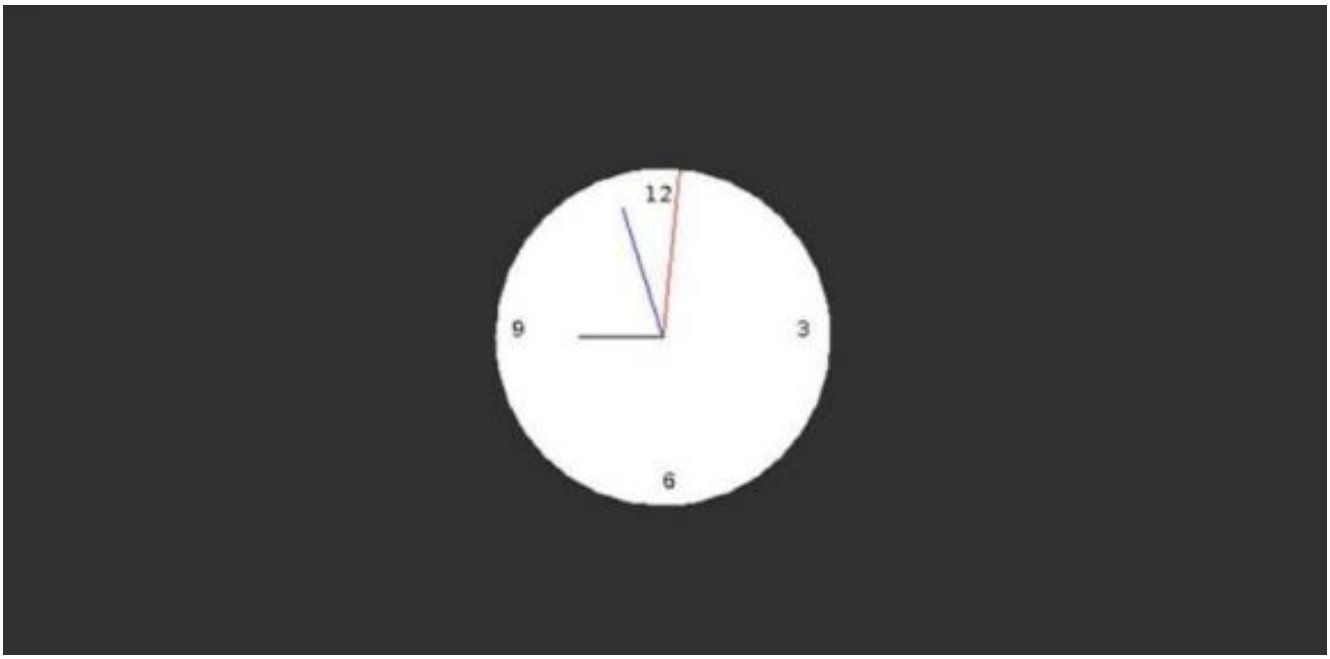
.html file

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Hello World</title>
  </head>
  <body>
    <applet code="prg_30_Applet1.class" width="300" height="300"></applet>
  </body>
</html>
```

Program – 31

```java
//Develop an Analog Clock using applet.

package LAB_07;

import java.applet.*;
import java.awt.*;
import java.util.*;

public class prg_31_Applet2 extends Applet {
    @Override
    public void init() {
        this.setSize(new Dimension(800, 400));
        setBackground(new Color(50, 50, 50));
        new Thread() {
            @Override
            public void run() {
                while (true) {
                    repaint();
                    delayAnimation();
                }
            }
        }.start();
    }

    private void delayAnimation() {
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void paint(Graphics g) {
        // Get the system time
        Calendar time = Calendar.getInstance();
        int hour = time.get(Calendar.HOUR_OF_DAY);
        int minute = time.get(Calendar.MINUTE);
        int second = time.get(Calendar.SECOND);
        // 12 hour format
        if (hour > 12) {
            hour -= 12;
        }
        // Draw clock body center at (400, 200)
        g.setColor(Color.white);
        g.fillOval(300, 100, 200, 200);
        // Labeling
        g.setColor(Color.black);
```

```
        g.drawString("12", 390, 120);
        g.drawString("9", 310, 200);
        g.drawString("6", 400, 290);
        g.drawString("3", 480, 200);
        // Declaring variables to be used
        double angle;
        int x, y;
        // Second hand's angle in Radian
        angle = Math.toRadians((15 - second) * 6);
        // Position of the second hand with length 100 unit
        x = (int) (Math.cos(angle) * 100);
        y = (int) (Math.sin(angle) * 100);
        // Red color second hand
        g.setColor(Color.red);
        g.drawLine(400, 200, 400 + x, 200 - y);
        // Minute hand's angle in Radian
        angle = Math.toRadians((15 - minute) * 6);
        // Position of the minute hand
        // with length 80 unit
        x = (int) (Math.cos(angle) * 80);
        y = (int) (Math.sin(angle) * 80);
        // blue color Minute hand
        g.setColor(Color.blue);
        g.drawLine(400, 200, 400 + x, 200 - y);
        // Hour hand's angle in Radian
        angle = Math.toRadians((15 - (hour * 5)) * 6);
        // Position of the hour hand
        // with length 50 unit
        x = (int) (Math.cos(angle) * 50);
        y = (int) (Math.sin(angle) * 50);
        // Black color hour hand
        g.setColor(Color.black);
        g.drawLine(400, 200, 400 + x, 200 - y);
    }
}
```



Lab Exercise - 8

Program – 32

```
//WAP to show Multi-Threading

package LAB_08;

class Base {
    int num;
    boolean ValueSet = false;

    public synchronized void Put(int n) {
        while (ValueSet == true) {
            try {
                wait();
            } catch (Exception e) {
            }
        }
        System.out.println("Put -->  num : " + n);
        this.num = n;
        ValueSet = true;
        notify();
    }

    public synchronized void Get() {
        while (ValueSet == false) {
            try {
                wait();
            } catch (Exception e) {
            }
        }
        System.out.println("Get -->  num : " + num);
        ValueSet = false;
        notify();
    }
}

class Producer implements Runnable {
    Base Obj;

    public Producer(Base b) {
        this.Obj = b;
        Thread T1 = new Thread(this, "Producer");
        T1.start();
    }

    @Override
```

```java
    public void run() {
        int i = 0;
        while (i <= 5) {
            Obj.Put(i++);
            try {
                Thread.sleep(1000);
            } catch (Exception e) {
                e.getStackTrace();
            }
        }
    }
}

class Consumer implements Runnable {
    Base Obj;

    public Consumer(Base b) {
        this.Obj = b;
        Thread T2 = new Thread(this, "Consumer");
        T2.start();
    }

    @Override
    public void run() {
        int i = 0;
        while (i <= 5) {
            Obj.Get();
            try {
                Thread.sleep(2000);
            } catch (Exception e) {
                e.getStackTrace();
            }
        }
    }
}

public class prg_32_MultiThread {
    public static void main(String args[]) {
        Base Obj = new Base();
        Producer P = new Producer(Obj);
        Consumer C = new Consumer(Obj);
    }
}
```

```
Put -->  num : 0
Get -->  num : 0
Put -->  num : 1
Get -->  num : 1
Put -->  num : 2
Get -->  num : 2
Put -->  num : 3
Get -->  num : 3
Put -->  num : 4
Get -->  num : 4
Put -->  num : 5
Get -->  num : 5
```

Program – 33

```
//WAP that executes to threads. One thread displays "An" after every 1000ms &
the other displays "B" after every 3000ms. Create the threads by executing the
thread class.

package LAB_08;

class First extends Thread {
    public void run() {
        for (int i = 0; i < 9; i++) {
            System.out.print("An    ");
            try {
                Thread.sleep(1000);
            } catch (Exception e) {
            }
        }
    }
}

class Second extends Thread {
    public void run() {
        for (int i = 0; i < 3; i++) {
            System.out.println("\nB    ");
            try {
                Thread.sleep(3000);
            } catch (Exception e) {
            }
        }
    }
}

public class prg_33_Thread1 {
    public static void main(String args[]) {
        System.out.println();
        First F = new First();
        Second S = new Second();
        F.start();
        try {
            Thread.sleep(3000);
        } catch (Exception e) {
        }
        S.start();
    }
}
```

```
An      An      An
B
An      An      An
B
An      An      An
B
```

Program – 34

```
//WAP & Create a class "salesperson" as a thread that
will display a salesperson name. Create a class "days" as
other thread that has array of 7 days. Call the instance
of "salesperson" in "days" and start both the threads.
Suspend the sales person on Sunday & resume on Wednesday.
We can only use suspend and resume methods
from the thread only.

package LAB_08;

class SalesPerson extends Thread {
    public String name;

    public SalesPerson(String n) {
        this.name = n;
    }

    public void run() {
        System.out.println("Sales_Person " + name + "
reporting for duty.");
        System.out.println();
    }
}

class Days extends Thread {
    String[] days = { "Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday" };
    SalesPerson salesPerson;

    public Days(SalesPerson salesPerson) {
        this.salesPerson = salesPerson;
    }

    public void run() {
        for (String day : days) {
            if (day.equals("Sunday")) {
                System.out.println("Sales_Person " +
salesPerson.name + " suspended shop on : " + day);
```

```
                // salesPerson.suspend();
            } else if (day.equals("Monday") ||
day.equals("Tuesday")) {
                continue;
            } else if (day.equals("Wednesday")) {
                System.out.println("Sales_Person " +
salesPerson.name + " resumed shop on : " + day);
            } else {
                System.out.println("Sales_Person " +
salesPerson.name + " continued shop on : " + day);
            }
        }
    }
}

public class prg_34_Thread2 {
    public static void main(String args[]) throws
Exception {
        System.out.println();
        SalesPerson SP = new SalesPerson("Amit");
        Days DP = new Days(SP);
        SP.start();
        DP.start();
        SP.join();
        DP.join();
        System.out.println();
    }
}
```

```
Sales_Person Amit reporting for duty.

Sales_Person Amit suspended shop on : Sunday
Sales_Person Amit resumed shop on : Wednesday
Sales_Person Amit continued shop on : Thursday
Sales_Person Amit continued shop on : Friday
Sales_Person Amit continued shop on : Saturday
```

Program – 35

```
//WAP that read & write in the file.

package LAB_08;

import java.io.*;
import java.util.*;

public class prg_35_FileHandling {
    public static void main(String args[]) {
        try {
            File myFile = new File("D:\\Admin\\B.Tech\\LAB-
WORK\\Java\\LAB_09\\prg_35_FileHandling.txt");
            myFile.createNewFile();
            FileWriter WriteFile = new FileWriter("D:\\Admin\\B.Tech\\LAB-
WORK\\Java\\LAB_09\\prg_35_FileHandling.txt");
            WriteFile.write("Hello, I Am Amit Singhal\nI Am 18 Years Old\n");
            WriteFile.write("Java is my Favourite Subject");
            WriteFile.close();
            Scanner sc = new Scanner(myFile);
            while (sc.hasNextLine()) {
                String line = sc.nextLine();
                System.out.println(line);
            }
            sc.close();
            myFile.delete();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

➔ TXT FILE

prg_35_FileHandling - Notepad

File Edit Format View Help

```
Hello, I Am Amit Singhal
I Am 18 Years Old
Java is my Favourite Subject
```

```
Hello, I Am Amit Singhal
I Am 18 Years Old
Java is my Favourite Subject
```