INDEX

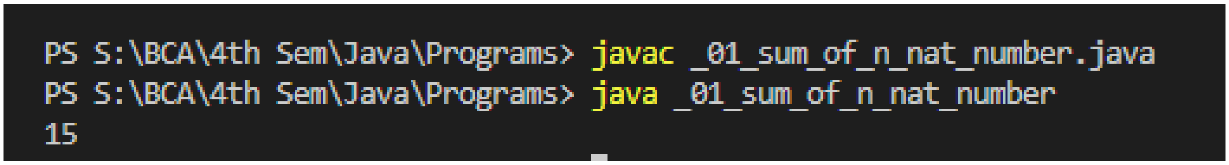
12.	Program for Command Line Arguments.	
13.	Program to split string to words:	
14.	Program to overload methods by parameters.	
15.	Program to overload methods on basis of type of parameters.	
16.	Default constructors Program.	
17.	Parameterized Constructor Program.	
18.	Overriding in Polymorphism Program.	
19.	Overloading in Polymorphism Program.	
20.	Single Inheritance	
21.	Multilevel Inheritance	
22.	Hierarchial Inheritance	
23.	Super Keyword in Inheritance.	
24.	Program of Abstract Class.	
25.	Program of Abstract Method.	

26.	Program for this keyword.	
27.	Program for interface in java.	
28.	Program of Inheriting Multiple Interface.	
29.	Exception Handling using try-catch	
30.	Arithmetic Exception using try-catch and finally block.	
31.	Multithreading using Thread	
32.	Multithreading using runnable	
33.	Program for awt button.	
34.	Program for AWT GUI extending Frame class.	
35.	Swing JButton Program	
36.	Swing JTextField Program	
37.	Swing JCheckBox Program	

1. Program to find Sum of n natural number.

```
// Sum of n natural number
class A{
public int sum(int n){
int sum = 0;
for(int i = 0; i<=n; i++){
sum += i;
}
return sum;
}
}

class _01_sum_of_n_nat_number{
public static void main(String[] args) {
A obj = new A();
int sum = obj.sum(5);

System.out.println(sum);
}

}
```

Output:



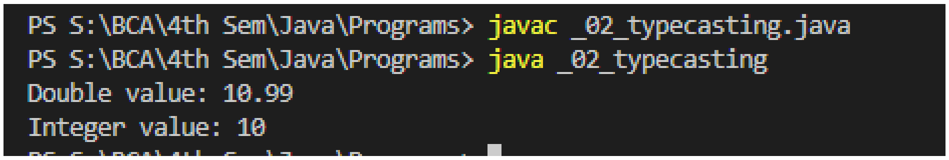
```
PS S:\BCA\4th Sem\Java\Programs> javac _01_sum_of_n_nat_number.java
PS S:\BCA\4th Sem\Java\Programs> java _01_sum_of_n_nat_number
15
```

Typecasting:

2. Program for Narrowing Typecasting.

```
// narrowing TypeCasting
public class _02_typecasting {


public static void main(String[] args) {
// double type variable
double num = 10.99;
System.out.println("Double value: " + num);


// double to int typecasting
int data = (int)num;
System.out.println("Integer value: " + data);
}
}
```

Output:

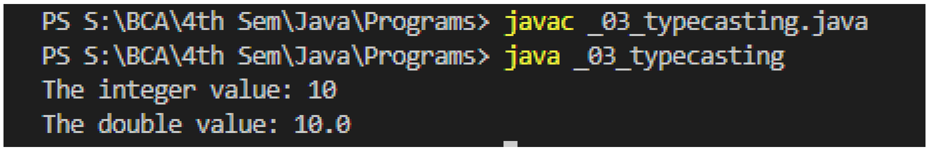


```
PS S:\BCA\4th Sem\Java\Programs> javac _02_typecasting.java
PS S:\BCA\4th Sem\Java\Programs> java _02_typecasting
Double value: 10.99
Integer value: 10
```

3. Program for Widening Typecasting.

```
// widening TypeCasting

public class _03_typecasting {

public static void main(String[] args) {

// Int variable

int num = 10;

System.out.println("The integer value: " + num);


// int to double conversion

double data = num;

System.out.println("The double value: " + data);

}

}
```

Output:

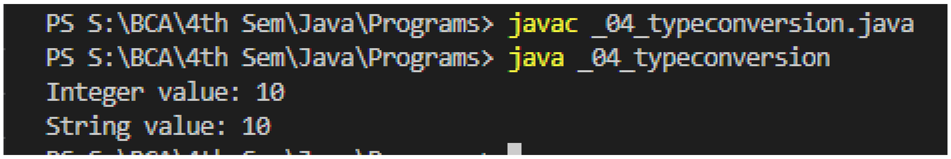


```
PS S:\BCA\4th Sem\Java\Programs> javac _03_typecasting.java
PS S:\BCA\4th Sem\Java\Programs> java _03_typecasting
The integer value: 10
The double value: 10.0
```

4. Program for typeconversion from int to string.

```
// int to string typeconversion
public class _04_typeconversion {
public static void main(String[] args) {
// int type variable
int num = 10;
System.out.println("Integer value: " + num);

// int to string type
String data = String.valueOf(num);
System.out.println("String value: " + data);
}
}
```

Output:



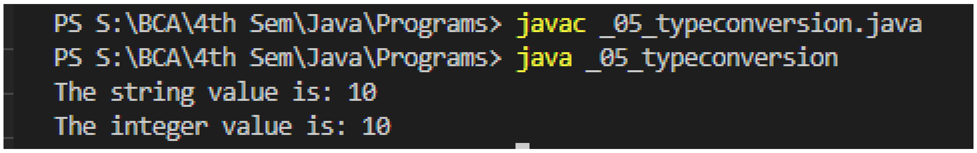
```
PS S:\BCA\4th Sem\Java\Programs> javac _04_typeconversion.java
PS S:\BCA\4th Sem\Java\Programs> java _04_typeconversion
Integer value: 10
String value: 10
```

5. Program for typeconversion from string to int.

```
// string to int typeconversion


public class _05_typeconversion {
public static void main(String[] args) {
```

```
// String type
String data = "10";
System.out.println("The string value is: " + data);

// string var to int
int num = Integer.parseInt(data);
System.out.println("The integer value is: " + num);
}
}
```

Output:



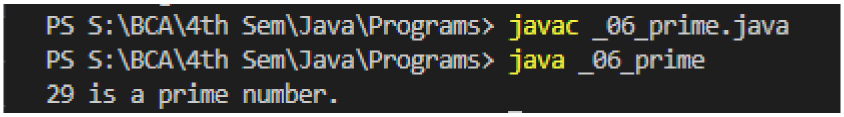
```
PS S:\BCA\4th Sem\Java\Programs> javac _05_typeconversion.java
PS S:\BCA\4th Sem\Java\Programs> java _05_typeconversion
The string value is: 10
The integer value is: 10
```

6. Program to check if the given number is prime or not.

```
// Program to check if number is prime or not
public class _06_prime {
static boolean isPrime(int num) {

boolean flag = false;
for (int i = 2; i <= num / 2; ++i) {
// condition for nonprime number
if (num % i == 0) {
```

```
flag = true;
break;
}
}

if (!flag)
return false;
else
return true;
}


public static void main(String[] args) {

int num = 29;

if (!isPrime(num))
System.out.println(num + " is a prime number.");
else
System.out.println(num + " is not a prime number.");
}
}
```

Output:

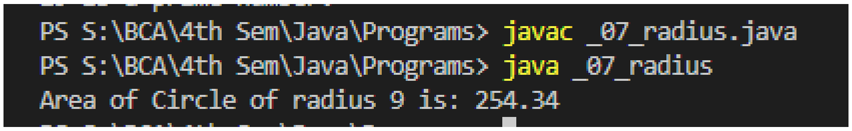


```
PS S:\BCA\4th Sem\Java\Programs> javac _06_prime.java
PS S:\BCA\4th Sem\Java\Programs> java _06_prime
29 is a prime number.
```

7. Program to calculate Area of the circle.

```
// Program to calculatre area of radius

public class _07_radius {

public static void main(String args[]){

final double pi = 3.14;

int r = 9;

double area = pi * r * r;

System.out.println("Area of Circle of radius " + r + " is: " + area);

}

}
```

Output:

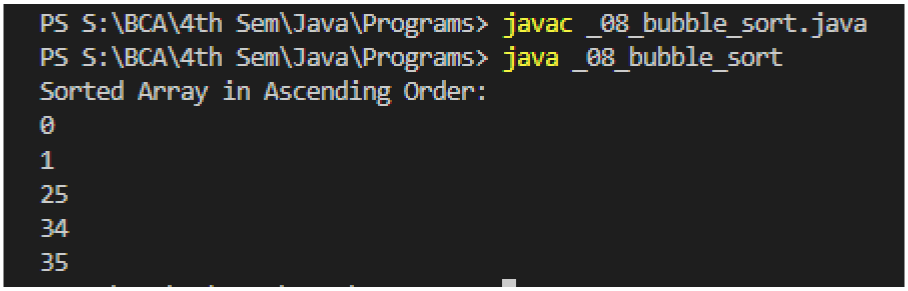


```
PS S:\BCA\4th Sem\Java\Programs> javac _07_radius.java
PS S:\BCA\4th Sem\Java\Programs> java _07_radius
Area of Circle of radius 9 is: 254.34
```

8. Sorting Program

```
import java.util.*;
public class _08_bubble_sort {

static void bubbleSort(int arr[]) {
int size = arr.length;

for (int i = 0; i < size - 1; i++)
for (int j = 0; j < size - i - 1; j++)
if (arr[j] > arr[j + 1]) {

int temp = arr[j];
arr[j] = arr[j + 1];
arr[j + 1] = temp;

}
}

public static void main(String args[]) {

int[] arr = { 25, 35, 34, 1, 0 };

// call method using class name
bubbleSort(arr);
System.out.println("Sorted Array in Ascending Order:");

for(int i = 0; i<arr.length ; i++){

System.out.println(arr[i] );

}

}

}
```

Output:

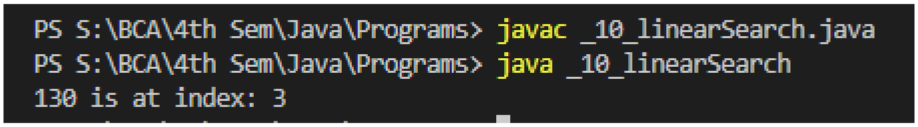


```
PS S:\BCA\4th Sem\Java\Programs> javac _08_bubble_sort.java
PS S:\BCA\4th Sem\Java\Programs> java _08_bubble_sort
Sorted Array in Ascending Order:
0
1
25
34
35
```

9. Searching Program

```
public class _10_linearSearch {

public static int linearSearch(int[] arr, int element){

for(int i=0;i<arr.length;i++){

if(arr[i] == element){

return i;

}

}

return -1;

}
public static void main(String a[]){

int[] a1= {100, 110,120, 130, 140, 150, 180};

int element = 130;

System.out.println(element +" is at index: " + linearSearch(a1, element));

}


}
```

Output:

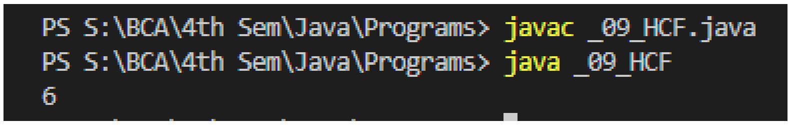


```
PS S:\BCA\4th Sem\Java\Programs> javac _10_linearSearch.java
PS S:\BCA\4th Sem\Java\Programs> java _10_linearSearch
130 is at index: 3
```

10. Program to find HCF of two numbers

```
public class _09_HCF {

static void HCF(int n1, int n2){

int hcf = 0;


for(int i = 1; i <= Math.min(n1, n2); i++){

if( n1%i == 0 && n2%i == 0){

hcf = i;

}

}
System.out.println(hcf);

}

public static void main(String[] args) {

HCF(12, 18);

}

}
```

Output:

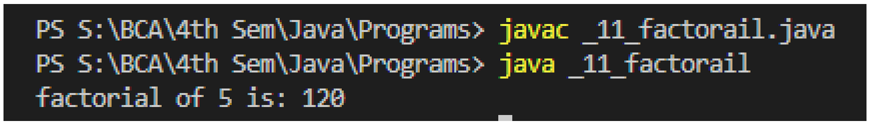


```
PS S:\BCA\4th Sem\Java\Programs> javac _09_HCF.java
PS S:\BCA\4th Sem\Java\Programs> java _09_HCF
6
```

11. Program to find the factorial of the number.

```
public class _11_factorail {

public static int factorial(int n){

if (n==1 || n==0){

return 1;

}

return n * factorial(n-1);

}

public static void main(String[] args) {

int n = 5;


System.out.println("factorial of " + n + " is: " + factorial(n));
}

}
```

Output:

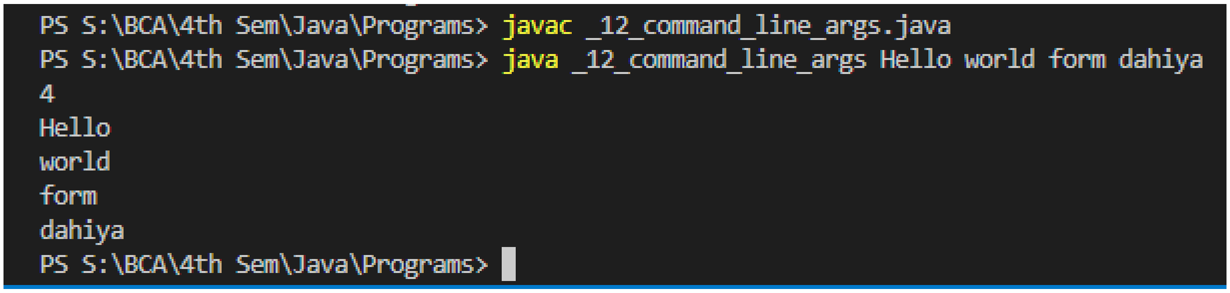


```
PS S:\BCA\4th Sem\Java\Programs> javac _11_factorail.java
PS S:\BCA\4th Sem\Java\Programs> java _11_factorail
factorial of 5 is: 120
```

12. Program for Command Line Arguments.

```
public class _12_command_line_args {

public static void main(String args[]){

System.out.println(args.length);

for(int i=0;i<args.length;i++)

System.out.println(args[i]);

}

}
```

OUTPUT:

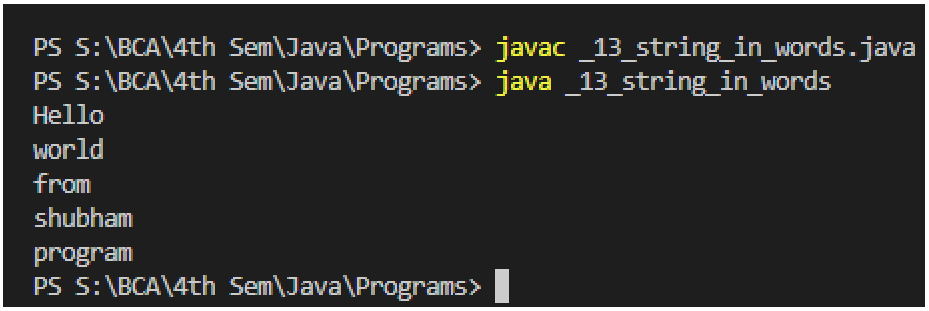


```
PS S:\BCA\4th Sem\Java\Programs> javac _12_command_line_args.java
PS S:\BCA\4th Sem\Java\Programs> java _12_command_line_args Hello world form dahiya
4
Hello
world
form
dahiya
PS S:\BCA\4th Sem\Java\Programs>
```

13. Program to split string to words:

```
public class _13_string_in_words{

static void splitString(String str){

int i =0;

for( ; i<str.length(); i++){

if(str.charAt(i) == ' '){

System.out.println();

continue;

}

System.out.print(str.charAt(i));

}

}

public static void main(String[] args){

String a = "Hello world from shubham program";

splitString(a);

}

}
```

OUTPUT:

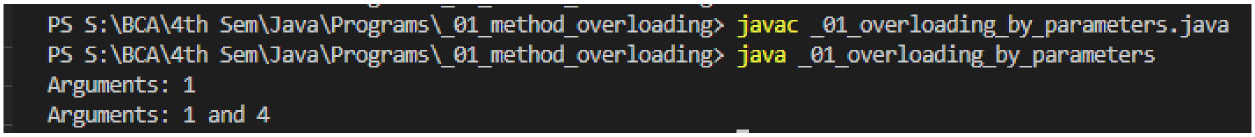


```
PS S:\BCA\4th Sem\Java\Programs> javac _13_string_in_words.java
PS S:\BCA\4th Sem\Java\Programs> java _13_string_in_words
Hello
world
from
shubham
program
PS S:\BCA\4th Sem\Java\Programs>
```

14. Program to overload methods by parameters.

```
//method overloading by parameters


public class _01_overloading_by_parameters {

private static void display(int a){

System.out.println("Arguments: " + a);

}


private static void display(int a, int b){

System.out.println("Arguments: " + a + " and " + b);

}


public static void main(String[] args) {

display(1);

display(1, 4);

}


}
```

Output:

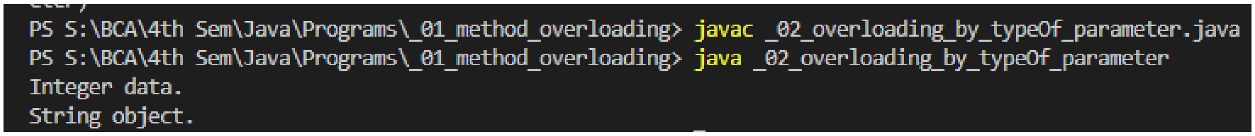


```
PS S:\BCA\4th Sem\Java\Programs\_01_method_overloading> javac _01_overloading_by_parameters.java
PS S:\BCA\4th Sem\Java\Programs\_01_method_overloading> java _01_overloading_by_parameters
Arguments: 1
Arguments: 1 and 4
```

15. Program to overload methods on basis of type of parameters.

```
public class _02_overloading_by_typeOf_parameter {
// this method accepts int
private static void display(int a){
System.out.println("Integer data.");
}

// this method accepts String object
private static void display(String a){
System.out.println("String object.");
}

public static void main(String[] args) {
display(1);
display("Hello");
}

}
```

Output:

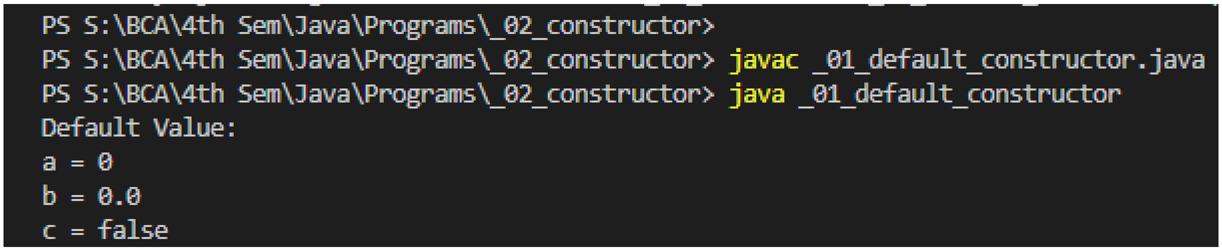


```
PS S:\BCA\4th Sem\Java\Programs\_01_method_overloading> javac _02_overloading_by_typeOf_parameter.java
PS S:\BCA\4th Sem\Java\Programs\_01_method_overloading> java _02_overloading_by_typeOf_parameter
Integer data.
String object.
```

16. Default constructors Program.

```
class Main{

int a;

double b;

boolean c;

}

public class _01_default_constructor {


public static void main(String[] args) {


// A default constructor is called

Main obj = new Main();


System.out.println("Default Value:");

System.out.println("a = " + obj.a);

System.out.println("b = " + obj.b);

System.out.println("c = " + obj.c);

}


}
```

Output:

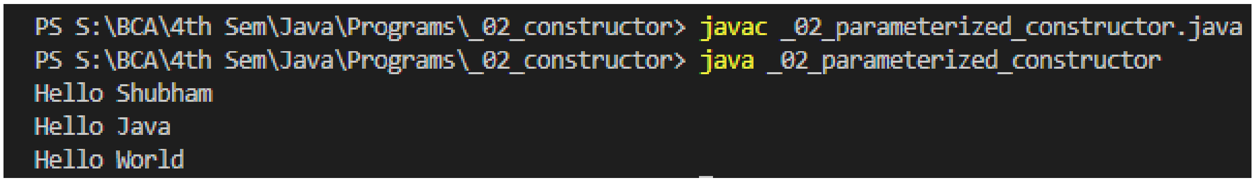


```
PS S:\BCA\4th Sem\Java\Programs\_02_constructor>
PS S:\BCA\4th Sem\Java\Programs\_02_constructor> javac _01_default_constructor.java
PS S:\BCA\4th Sem\Java\Programs\_02_constructor> java _01_default_constructor
Default Value:
a = 0
b = 0.0
c = false
```

17. Parameterized Constructor Program.

```
class Main {

String name;

// constructor accepting single value
Main(String n) {
name = n;
System.out.println("Hello " + name);
}
}

public class _02_parameterized_constructor {

public static void main(String[] args) {

// call constructor by passing a single value
Main obj1 = new Main("Shubham ");
Main obj2 = new Main("Java");
Main obj3 = new Main("World");
}
}
```

Output:

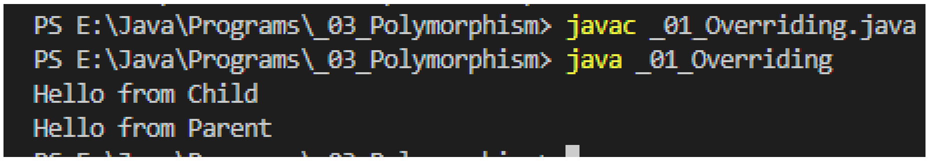


```
PS S:\BCA\4th Sem\Java\Programs\_02_constructor> javac _02_parameterized_constructor.java
PS S:\BCA\4th Sem\Java\Programs\_02_constructor> java _02_parameterized_constructor
Hello Shubham
Hello Java
Hello World
```

18. Overriding in Polymorphism Program.

```
class Parent {
public void sayHello() {
System.out.println("Hello from Parent");
}
}

class Child extends Parent {
@Override
public void sayHello() {
System.out.println("Hello from Child");
}
}

public class _01_Overriding {
public static void main(String[] args) {

// create an object of Child class
Child j1 = new Child();
j1.sayHello();

// create an object of Parent class
Parent l1 = new Parent();
l1.sayHello();
}
}
```

Output:

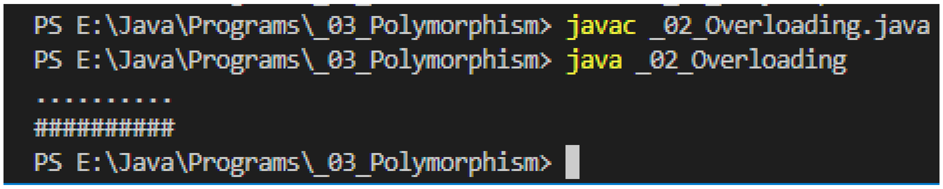


```
PS E:\Java\Programs\_03_Polymorphism> javac _01_Overriding.java
PS E:\Java\Programs\_03_Polymorphism> java _01_Overriding
Hello from Child
Hello from Parent
```

19. Overloading in Polymorphism Program.

```
class Pattern {

// method without parameter
public void display() {
for (int i = 0; i < 10; i++) {
System.out.print(".");
}
}

// method with single parameter
public void display(char symbol) {
for (int i = 0; i < 10; i++) {
System.out.print(symbol);
}
}
}

public class _02_Overloading {
public static void main(String[] args) {
Pattern d1 = new Pattern();

d1.display();
System.out.println();
d1.display('#');
}
}
```

Output:



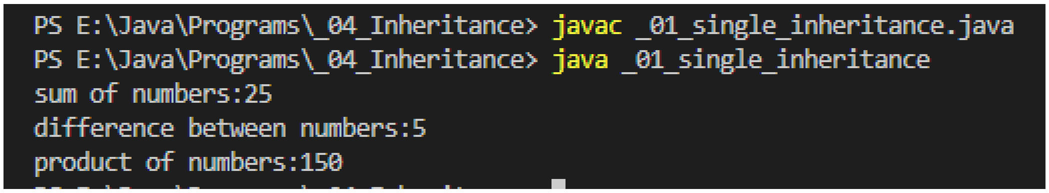
```
PS E:\Java\Programs\_03_Polymorphism> javac _02_Overloading.java
PS E:\Java\Programs\_03_Polymorphism> java _02_Overloading
..........
##########
PS E:\Java\Programs\_03_Polymorphism>
```

20. Single Inheritance

```
class Add_Sub {

int result;


public void addition(int a, int b) {

result = a + b;

System.out.println("sum of numbers:" + result);

}


public void Subtraction(int a, int b) {

result = a - b;

System.out.println("difference between numbers:" + result);

}

}


//inherited class

class Add_Sub_Mul extends Add_Sub {

public void multiplication(int a, int b) {
```

```
result = a * b;

System.out.println("product of numbers:" + result);

}

}


public class _01_single_inheritance {


public static void main(String args[]) {

int a = 15, b = 10;

Add_Sub_Mul obj = new Add_Sub_Mul();

obj.addition(a, b);

obj.Subtraction(a, b);

obj.multiplication(a, b);

}

}
```

<u>Output:</u>

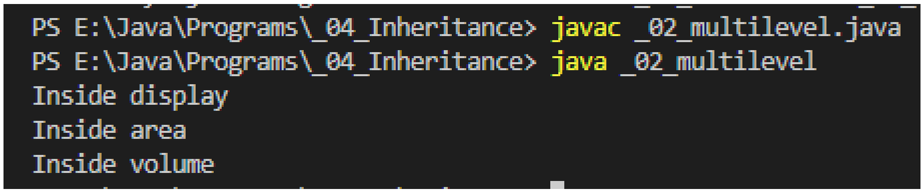


```
PS E:\Java\Programs\_04_Inheritance> javac _01_single_inheritance.java
PS E:\Java\Programs\_04_Inheritance> java _01_single_inheritance
sum of numbers:25
difference between numbers:5
product of numbers:150
```

21. Multilevel Inheritance

```
class Shape {
public void display() {
System.out.println("Inside display");
}
}
class Rectangle extends Shape {
public void area() {
System.out.println("Inside area");
}
}
class Cube extends Rectangle {
public void volume() {
System.out.println("Inside volume");
}
}

public class _02_multilevel {
public static void main(String[] arguments) {
Cube cube = new Cube();
cube.display();
cube.area();
cube.volume();
}
}
```

Output:



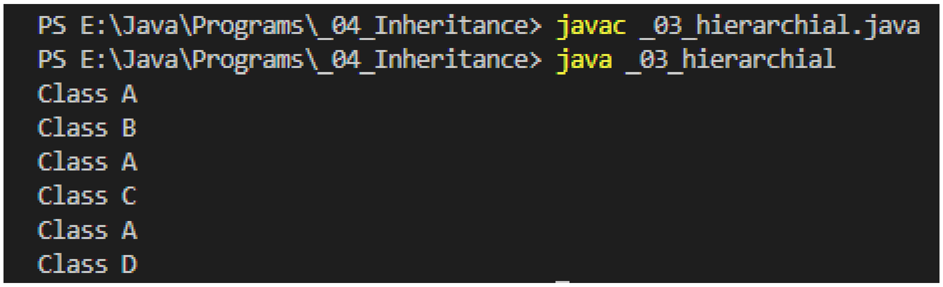
```
PS E:\Java\Programs\_04_Inheritance> javac _02_multilevel.java
PS E:\Java\Programs\_04_Inheritance> java _02_multilevel
Inside display
Inside area
Inside volume
```

22. Hierarchial Inheritance

```
class A {
   public void printA() { System.out.println("Class A"); }
}

class B extends A {
   public void printB() { System.out.println("Class B"); }
}

class C extends A {
   public void printC() { System.out.println("Class C"); }
}
```

```
class D extends A {
    public void printD() { System.out.println("Class D"); }
}

public class _03_hierarchial {
    public static void main(String[] args)
    {
        B objB = new B();
        objB.printA();
        objB.printB();

        C objC = new C();
        objC.printA();
        objC.printC();

        D objD = new D();
        objD.printA();
        objD.printD();
    }
}
```

Output:



```
PS E:\Java\Programs\_04_Inheritance> javac _03_hierarchial.java
PS E:\Java\Programs\_04_Inheritance> java _03_hierarchial
Class A
Class B
Class A
Class C
Class A
Class D
```

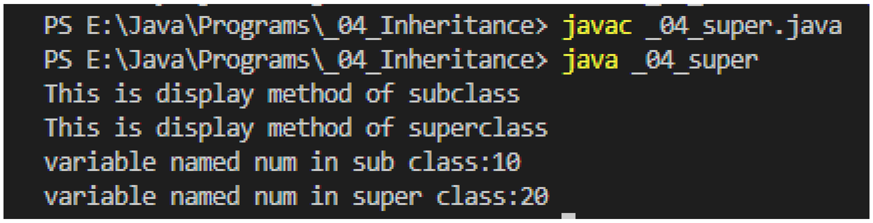
23. Super Keyword in Inheritance.

```
class Super_class {

int num = 20;


// display method of superclass

public void display() {

System.out.println("This is display method of superclass");

}

}


class Sub_class extends Super_class {

int num = 10;


// display method of sub class

public void display() {

System.out.println("This is display method of subclass");
```

```
}

public void my_method() {
// Instantiating subclass
Sub_class sub = new Sub_class();

// Invoking the display() method of sub class
sub.display();

// Invoking the display() method of superclass
super.display();

// printing the value of variable num of subclass
System.out.println("variable named num in sub class:"+ sub.num);

// printing the value of variable num of superclass
System.out.println("variable named num in super class:"+ super.num);
}
}

public class _04_super {
public static void main(String args[]) {
Sub_class obj = new Sub_class();
obj.my_method();
```

```
}

}
```

Output:

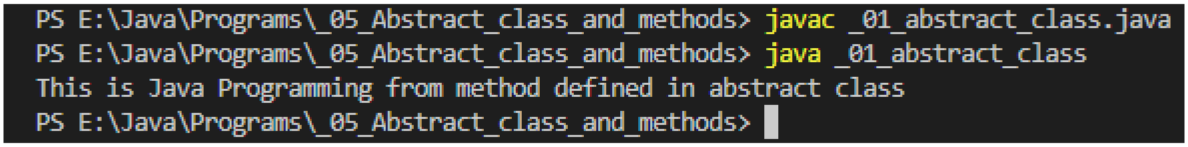


```
PS E:\Java\Programs\_04_Inheritance> javac _04_super.java
PS E:\Java\Programs\_04_Inheritance> java _04_super
This is display method of subclass
This is display method of superclass
variable named num in sub class:10
variable named num in super class:20
```

24. Program of Abstract Class.

```
abstract class abstractClass {


// method of abstract class

public void display() {

System.out.println("This is Java Programming from method defined in abstract class");

}

}


class subclass extends abstractClass {


public subclass() {

display();

}
}


public class _01_abstract_class {

public static void main(String[] args) {

subclass obj = new subclass();

}

}
```

Output:

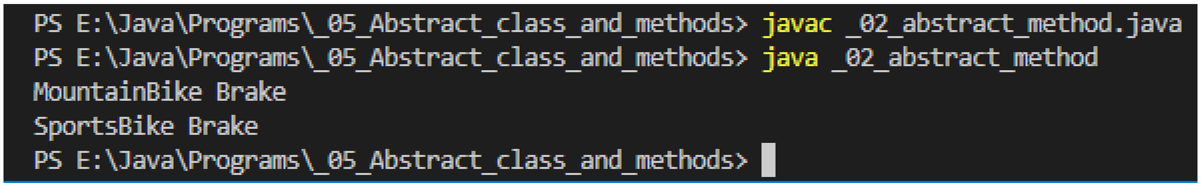


```
PS E:\Java\Programs\_05_Abstract_class_and_methods> javac _01_abstract_class.java
PS E:\Java\Programs\_05_Abstract_class_and_methods> java _01_abstract_class
This is Java Programming from method defined in abstract class
PS E:\Java\Programs\_05_Abstract_class_and_methods> 
```

25. Program of Abstract Method.

```
abstract class MotorBike {

abstract void brake();

}


class SportsBike extends MotorBike {


// implementation of abstract method

public void brake() {

System.out.println("SportsBike Brake");

}
}

class MountainBike extends MotorBike {

// implementation of abstract method
public void brake() {
System.out.println("MountainBike Brake");
}
}

public class _02_abstract_method {
public static void main(String[] args) {
MountainBike obj1 = new MountainBike();
obj1.brake();
SportsBike obj2 = new SportsBike();
obj2.brake();
}
}
```

Output:

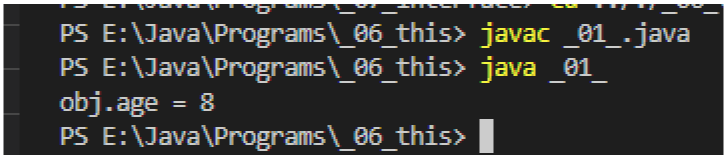


```
PS E:\Java\Programs\_05_Abstract_class_and_methods> javac _02_abstract_method.java
PS E:\Java\Programs\_05_Abstract_class_and_methods> java _02_abstract_method
MountainBike Brake
SportsBike Brake
PS E:\Java\Programs\_05_Abstract_class_and_methods>
```

26. Program for this keyword.

```
class C1 {


int age;

C1(int age){

this.age = age;

}

}

public class _01_ {

public static void main(String[] args) {

C1 obj = new C1(8);

System.out.println("obj.age = " + obj.age);

}

}
```

Output:

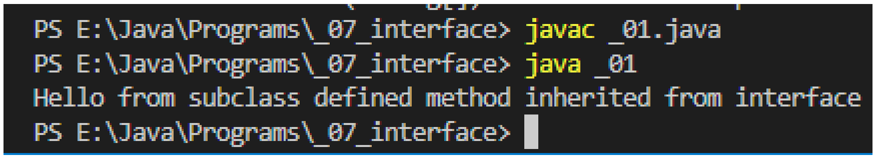


```
PS E:\Java\Programs\_06_this> javac _01_.java
PS E:\Java\Programs\_06_this> java _01_
obj.age = 8
PS E:\Java\Programs\_06_this>
```

27. Program for interface in java.

```
interface printable{

void print();

}

class Subclass implements printable{

public void print(){System.out.println("Hello from subclass defined method inherited from interface");}

}

public class _01 {

public static void main(String args[]){

Subclass obj = new Subclass();

obj.print();

}

}
```

Output:



```
PS E:\Java\Programs\_07_interface> javac _01.java
PS E:\Java\Programs\_07_interface> java _01
Hello from subclass defined method inherited from interface
PS E:\Java\Programs\_07_interface>
```

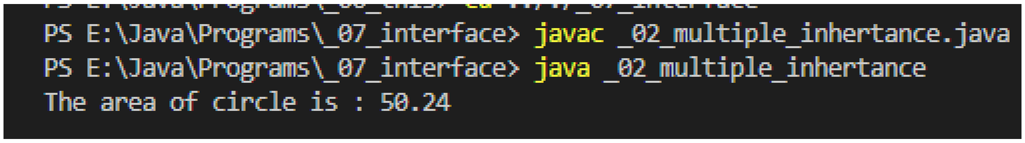
28. Program of Inheriting Multiple Interface.

```
//calculate area of the circle using Interface class.
```

```
interface get{                  //creating interface get and declaring a method inside it.

void get_r(int r);

}


interface ar{                   //creating interface ar and declaring a method inside it.

void area();

}


class calc implements get,ar{       //importing interface using "implements"
keyword.

final float pi=3.14f;


float r,ar;


public void get_r(int r){

this.r= r;                          //for taking input from user.

}

public void area(){

ar=pi*r*r;                               //calculating area using formula.

System.out.println("The area of circle is : " + ar);

}

}


class _02_multiple_inhertance{

public static void main(String args[]){
```

```
calc obj=new calc();           //creating object using derived class.

obj.get_r(4);                  //calling get_r function using object.

obj.area();                    //calling area function using object.


System.out.println("");


get obj1=new calc();           //creating object using the interface class.

obj1.get_r(6);                 //calling get_r function using object but we can't call the
other methods available in the calc because get interface has only get_r function.

}

}
```

<u>Output:</u>

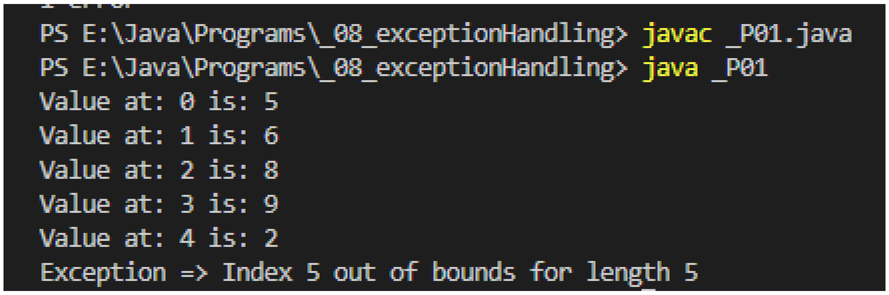


```
PS E:\Java\Programs\_06_this> cd ..\..\_07_interface
PS E:\Java\Programs\_07_interface> javac _02_multiple_inhertance.java
PS E:\Java\Programs\_07_interface> java _02_multiple_inhertance
The area of circle is : 50.24
```

29. Exception Handling using try-catch

```
public class _P01 {

public static void main(String[] args) {

int[] arr= {5, 6, 8, 9, 2};

try {

for (int i = 0; i < 7; i++) {

System.out.println("Value at: " + i + " is: " + arr[i]);
}

}

catch (Exception e) {

System.out.println("Exception => " + e.getMessage());

}

}

}
```

Output:

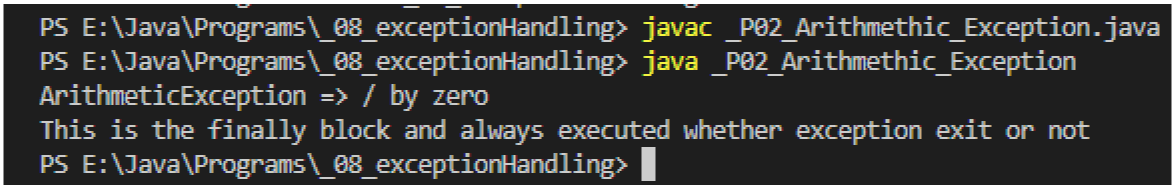


```
PS E:\Java\Programs\_08_exceptionHandling> javac _P01.java
PS E:\Java\Programs\_08_exceptionHandling> java _P01
Value at: 0 is: 5
Value at: 1 is: 6
Value at: 2 is: 8
Value at: 3 is: 9
Value at: 4 is: 2
Exception => Index 5 out of bounds for length 5
```

30. Arithmetic Exception using try-catch and finally block.

```
public class _P02_Arithmethic_Exception {

public static void main(String[] args) {

try {

// code that generates exception

int result = 5 / 0;

}
catch (ArithmeticException e) {

System.out.println("ArithmeticException => " + e.getMessage());

}


finally {

System.out.println("This is the finally block and always executed whether exception exit or not");

}

}

}
```

Output:

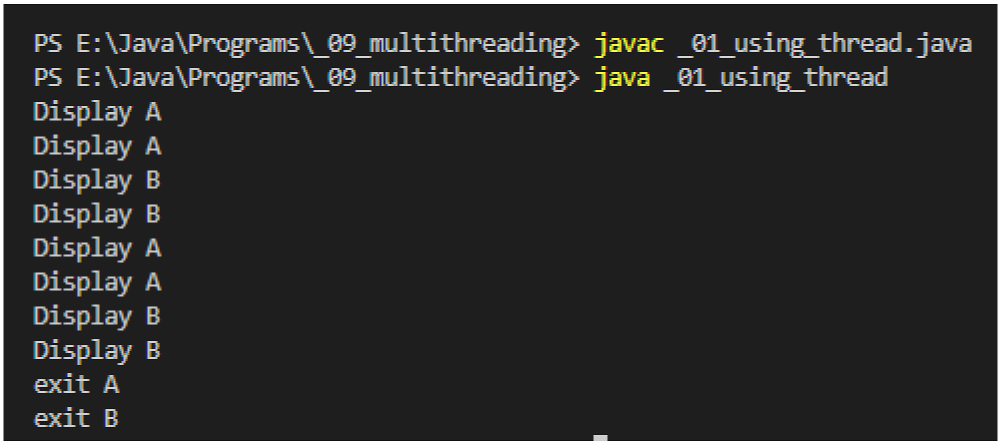


```
PS E:\Java\Programs\_08_exceptionHandling> javac _P02_Arithmethic_Exception.java
PS E:\Java\Programs\_08_exceptionHandling> java _P02_Arithmethic_Exception
ArithmeticException => / by zero
This is the finally block and always executed whether exception exit or not
PS E:\Java\Programs\_08_exceptionHandling>
```

31. Multithreading using Thread

```
class A extends Thread{

public void run(){

for(int i = 1; i<5; i++){

System.out.println("Display A");

}

System.out.println("exit A");
}
}

class B extends Thread{
public void run(){
for(int i = 1; i<5; i++){
System.out.println("Display B");
}
System.out.println("exit B");

}
}

class _01_using_thread{
public static void main(String[] args){

A obj1 = new A();
B obj2 = new B();

obj1.start();
obj2.start();
}

}
```

Output:

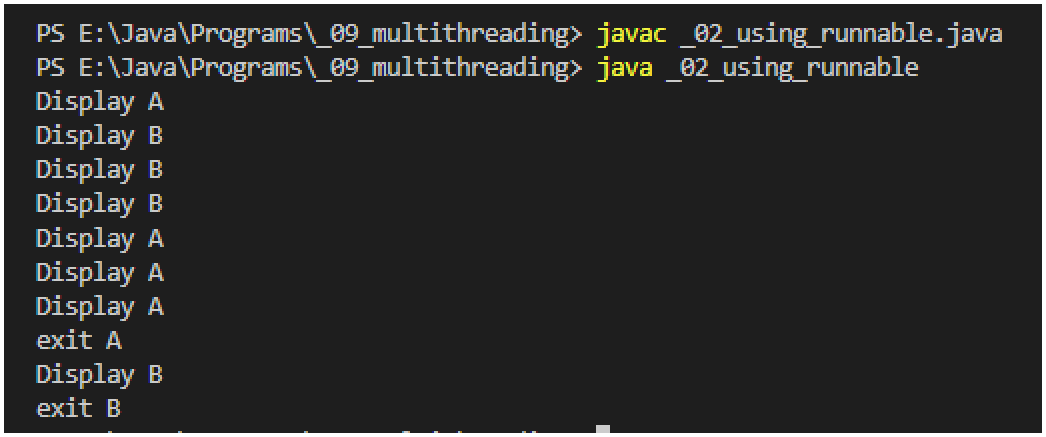


```
PS E:\Java\Programs\_09_multithreading> javac _01_using_thread.java
PS E:\Java\Programs\_09_multithreading> java _01_using_thread
Display A
Display A
Display B
Display B
Display A
Display A
Display B
Display B
exit A
exit B
```

32. Multithreading using runnable

```
class A implements Runnable{
public void run(){
for(int i = 1; i<5; i++){
System.out.println("Display A");
}
System.out.println("exit A");

}
}

class B implements Runnable{
public void run(){
for(int i = 1; i<5; i++){
System.out.println("Display B");
}
System.out.println("exit B");
}
}


class _02_using_runnable{
public static void main(String[] args){


Thread obj1 = new Thread (new A());
Thread obj2 = new Thread(new B());


obj1.start();
obj2.start();
}


}
```

<u>Output:</u>

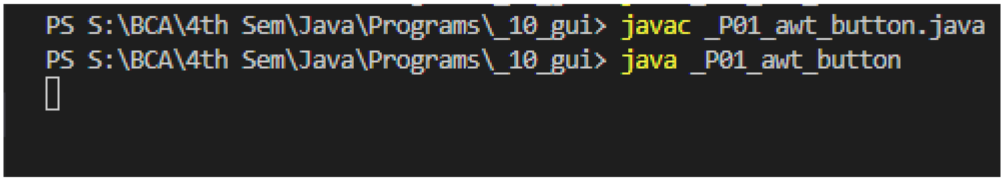


```
PS E:\Java\Programs\_09_multithreading> javac _02_using_runnable.java
PS E:\Java\Programs\_09_multithreading> java _02_using_runnable
Display A
Display B
Display B
Display B
Display A
Display A
Display A
exit A
Display B
exit B
```

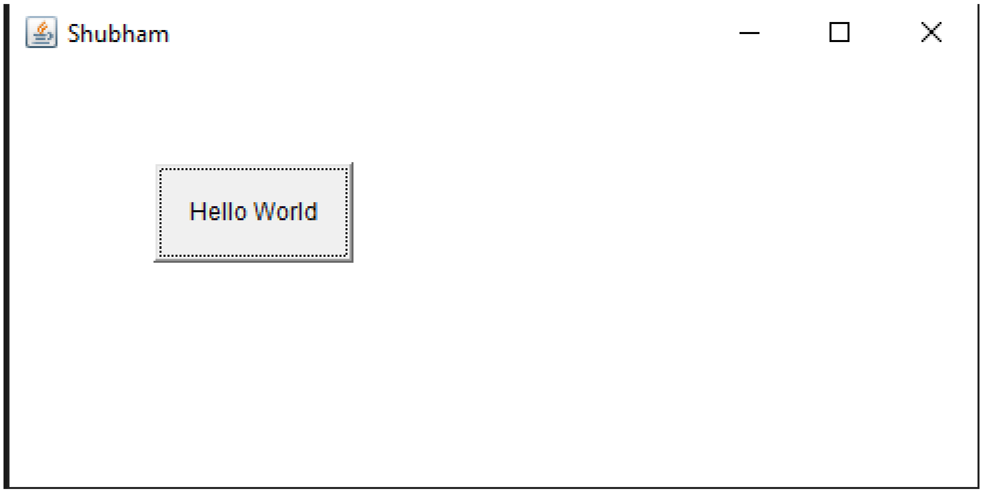
33. Program for awt button.

```
import java.awt.*;


class awtButton{
public awtButton()
{
Frame f = new Frame();
Button btn=new Button("Hello World");
btn.setBounds(80, 80, 100, 50);
//adding Button.
f.add(btn);
//setting size of frame.
f.setSize(800, 250);
// frame title
f.setTitle("Shubham");
f.setLayout(null);
f.setVisible(true);                //set frame visibility true.
}
}
public class _P01_awt_button {
public static void main(String[] args) {
awtButton awt = new awtButton();    //creating frame.
}
}
```

OUTPUT:



```
PS S:\BCA\4th Sem\Java\Programs\_10_gui> javac _P01_awt_button.java
PS S:\BCA\4th Sem\Java\Programs\_10_gui> java _P01_awt_button
```

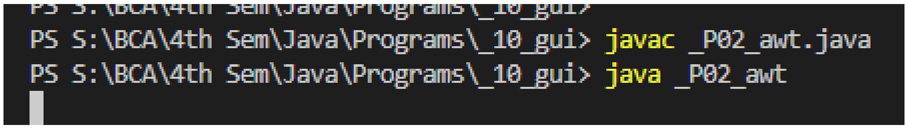


34. Program for AWT GUI extending Frame class.

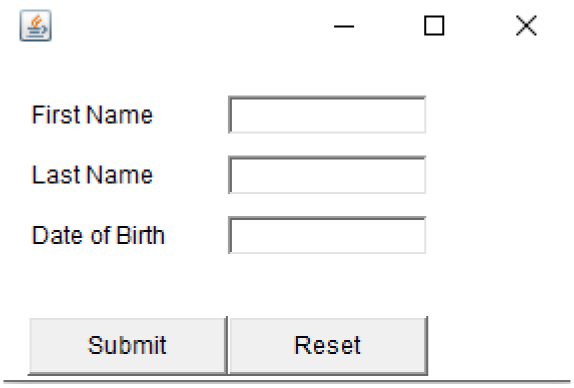
```
import java.awt.*;

class Awt extends Frame{

Awt(){

Label f_name = new Label("First Name");

f_name.setBounds(20, 50, 80, 20);


Label l_name = new Label("Last Name");

l_name.setBounds(20, 80, 80, 20);


Label dob = new Label("Date of Birth");
```

```
dob.setBounds(20, 110, 80, 20);

TextField f_ameTF = new TextField();
f_ameTF.setBounds(120, 50, 100, 20);

TextField l_NameTF = new TextField();
l_NameTF.setBounds(120, 80, 100, 20);

TextField dobTF = new TextField();
dobTF.setBounds(120, 110, 100, 20);

Button sbmt = new Button("Submit");
sbmt.setBounds(20, 160, 100, 30);

Button reset = new Button("Reset");
reset.setBounds(120,160,100,30);

add(f_name);
add(l_name);
add(dob);
add(f_ameTF);
add(l_NameTF);
add(dobTF);
add(sbmt);
add(reset);


setSize(300,200);
setLayout(null);
setVisible(true);
}
}
public class _P02_awt extends Frame {
public static void main(String[] args) {
Awt awt = new Awt();
}
}
```

OUTPUT:



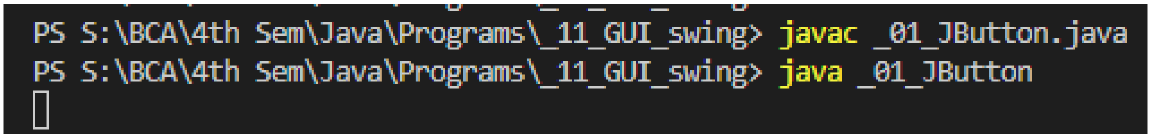
```
PS S:\BCA\4th Sem\Java\Programs\_10_gui>
PS S:\BCA\4th Sem\Java\Programs\_10_gui> javac _P02_awt.java
PS S:\BCA\4th Sem\Java\Programs\_10_gui> java _P02_awt
```



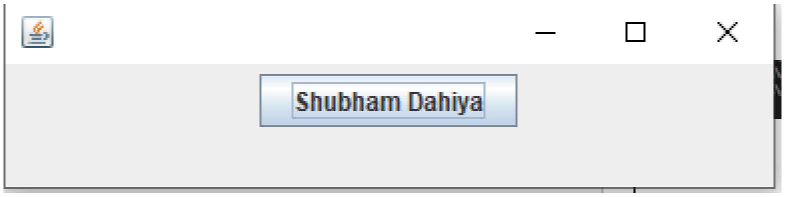
35. Swing JButton Program

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class _01_JButton extends JFrame
{
_01_JButton()
{
// Button.
JButton bt1 = new JButton("Shubham Dahiya");
//adding close operation --> Monica mam way
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new FlowLayout());        //setting layout using FlowLayout object
setSize(400, 100);        // size of Jframe
//adding button to frame
add(bt1);

setVisible(true);
}
public static void main(String[] args)
{
new _01_JButton();
}
}
```

OUTPUT:



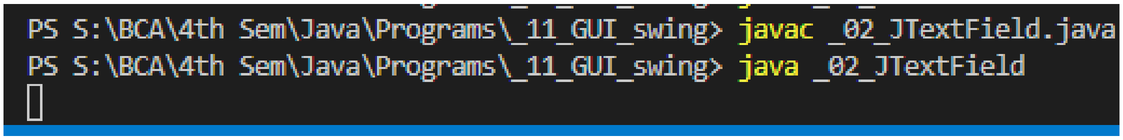
```
PS S:\BCA\4th Sem\Java\Programs\_11_GUI_swing> javac _01_JButton.java
PS S:\BCA\4th Sem\Java\Programs\_11_GUI_swing> java _01_JButton
```



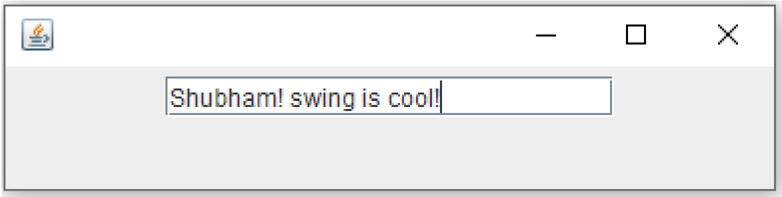
36. Swing JTextField Program

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class _02_JTextField extends JFrame
{
public _02_JTextField()
{
//creating JTextField.
JTextField jtf = new JTextField(20);
//adding JTextField to frame.
add(jtf);
setLayout(new FlowLayout());
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(400, 100);
setVisible(true);
}
public static void main(String[] args)
{
new _02_JTextField();
}
}
```

OUTPUT:



```
PS S:\BCA\4th Sem\Java\Programs\_11_GUI_swing> javac _02_JTextField.java
PS S:\BCA\4th Sem\Java\Programs\_11_GUI_swing> java _02_JTextField
```

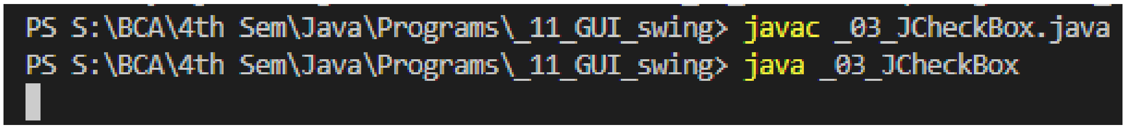


Shubham! swing is cool!

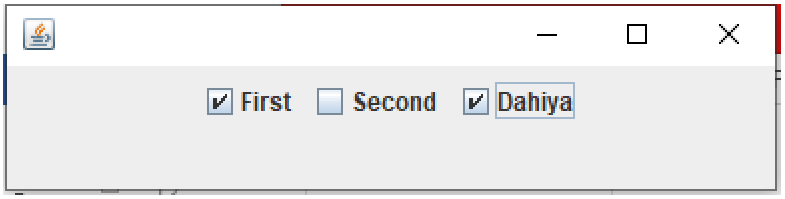
37. Swing JCheckBox Program

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class _03_JCheckBox extends JFrame
{
public _03_JCheckBox()
{
//creating JCheckBox.
JCheckBox jcb = new JCheckBox("First");
//adding JCheckBox to frame.
add(jcb);

jcb = new JCheckBox("Second");
add(jcb);

jcb = new JCheckBox("Dahiya");
add(jcb);

setLayout(new FlowLayout());
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(400, 100);
setVisible(true);
}
public static void main(String[] args)
{
new _03_JCheckBox();
}
}
```

OUTPUT:



```
PS S:\BCA\4th Sem\Java\Programs\_11_GUI_swing> javac _03_JCheckBox.java
PS S:\BCA\4th Sem\Java\Programs\_11_GUI_swing> java _03_JCheckBox
```



☑ First ☐ Second ☑ Dahiya