

Table of Contents

|  |  |  |
| --- | --- | --- |
| **Program No.** | **Program Name** | **Page No** |
| 13 | Create a package **pack1** having one class **C1** and one interface **I1**. Class **C1** has two methods **int sum(int, int)** and **int sub(int, int)**. The I1 has one method **int division(int, int)**. Create another package **pack2** having class **C2**. Reuse **C1** and **I1** in **C2** and show the results.  **Note: Use appropriate Access Modifiers as required.** | 35 |
| 14 | Write a program to divide two numbers with proper exception handlers. | 37 |
| 15 | Create LowBalanceException that occurs when user tries to withdraw some amount that is greater than his current bank balance. To withdraw you have to write a **void**  **withdrawal(int amount)** method. | 39 |
| 16 | Write a program that reads from a text file byte by byte and writes in some another file. Write this program in an efficient way. | 41 |
| 17 | Write a program that reads from a text file char by char and  writes in some another file. Write this program in an efficient way. | 43 |
| 18 | Write a program that reads from a text file line by line and writes on console. | 45 |
| 19 | Write a program that take your name from keyboard and writes in some text file. | 47 |
| 20 | Write a multithreaded program where three threads are there and printing the numbers from 1 to 10 concurrently. | 49 |
| 21 | Write a program to set and get the name of threads also set and get the priority of threads. | 51 |

|  |  |  |
| --- | --- | --- |
| 22 | Write a class Display having void wish(String name) methods that wishes hello to given string name. Between printing hello and provided string name apply delay of 500 milliseconds. Suppose multiple threads are there and they are trying to access this wish() method concurrently on  **same object** then irregular output will be there. Write this application in such a way so that output becomes regular. | 53 |
| 23 | Write a class Display having **synchronized void wish(String)** methods that wishes hello to given string name. Between printing hello and provided string name apply delay of 500 milliseconds. Suppose multiple threads are there and they are trying to access this wish() method concurrently on **different objects** then irregular output will be there. Write this application in such a way so that output  becomes regular. | 56 |
| 24 | Write a class Customer having **balance** as a property and **void withdrawal(int amount)**, and **void deposit(int amount)** as instance methods. There are two threads, the first thread wants to withdrawal some amount and second thread wants to deposit some amount. Apply inter thread communication where, if withdrawal amount is higher than current balance then first thread will wait for second thread to deposit then resume the withdrawal. | 59 |
| 25 | Create a GUI for student’s information system. A GUI that asks all the relevant information’s related to a student. | 62 |
| 26 | Create a canvas having smiley face. | 64 |
| 27 | Write a JFrame having three textfields. The first two textfields refers to two numbers on which sum or subtraction will happen. The third textfield will show the result. There are two buttons “SUM” and “SUBTRACTION”. Once user will click on sum or subtraction  buttons then the corresponding result will be displayed in result field. | 66 |
| 28 | Write a Java program that interacts with database. It enables to-   1. Inserts the student name and roll number to database. 2. Fetch records from table 3. Modify the records 4. Delete the records | 69 |

**Program 13**

## Source Code:

package pack1; public class C1{

public int sum( int x, int y ){ return x + y;

}

public int sub( int x, int y ){ return x - y;

}

}

package pack1; public interface I1{

public int div( int x, int y );

}

package pack2; import pack1.\*;

class C2 implements I1 {

public static void main(String[] args) { C1 obj = new C1();

C2 ref = new C2();

System.out.println( "sum: " + obj.sum( 4, 5 ) ); System.out.println( "sub: " + obj.sub( 9, 5 ) ); System.out.println( "div: " + ref.div( 9, 3 ) );

}

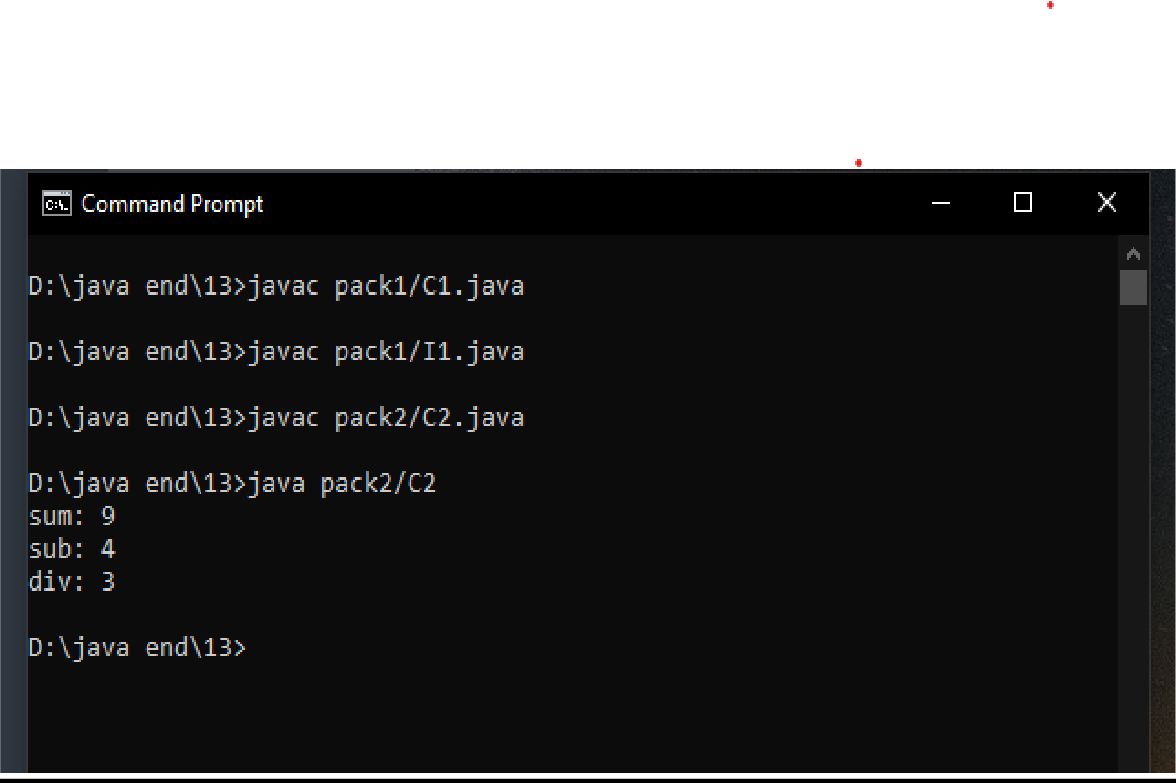
public int div( int x, int y ){

return x / y;

}

}

## OUTPUT



**Program 14**

## Source Code:

import java.util.Scanner; class ExH{

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

int a = sc.nextInt(); int b = sc.nextInt();

try{

System.out.println( a / b );

}

catch( Exception e ){

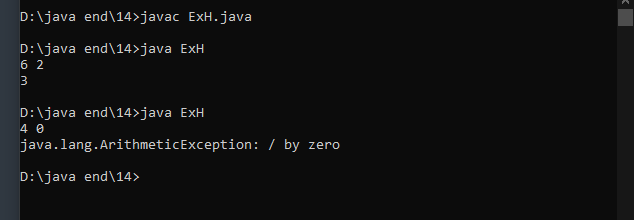
System.out.println( e );

}

}

}

## OUTPUT



**Source Code:**

**Program 15**

class UserDefinedException extends Exception {

UserDefinedException( String s ){ super( s );

}

}

class Bank{

int bal;

void withdraw( int amt ) throws UserDefinedException { if( amt > bal )

throw new UserDefinedException( "Low Balance Exception Error" );

else

}

}

bal -= amt;

class LowBalanceException{

public static void main(String[] args) { Bank bk = new Bank();

bk.bal = 1000; try{

bk.withdraw( 2000 );

System.out.println( "Updated Balance: " + bk.bal );

}

catch( UserDefinedException e ){

System.out.println( e );

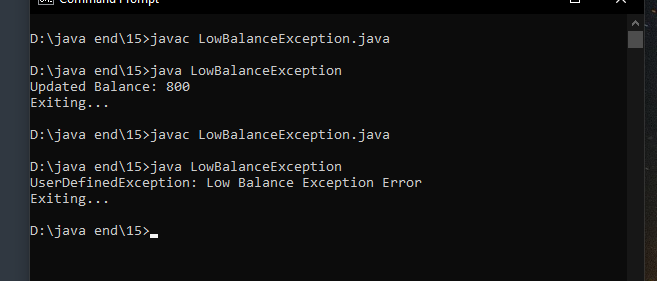
}

System.out.println( "Exiting..." );

}

}

## OUTPUT



**Source Code:**

**Program 16**

import java.io.FileInputStream; import java.io.FileWriter; import java.io.IOException;

class ByByte{

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

// read() => reads data from file char by char

FileInputStream fin = new FileInputStream ( "New Text Document.txt"

);

FileWriter fw = new FileWriter( "AnotherDocumnet.txt" ); byte by;

// fw.write((byte) fin.read() );

while( ( by = (byte) fin.read()) != -1 ){ fw.write( by );

}

fin.close();

fw.close();

}

catch( IOException e ){

System.out.println( e );

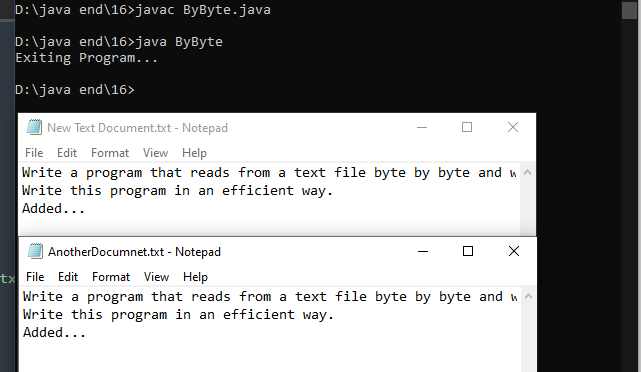
}

System.out.println( "Exiting Program..." );

}

}

## OUTPUT



**Program 17**

## Source Code:

import java.io.FileReader; import java.io.FileWriter; import java.io.IOException;

class readFile{

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

// read() => reads data from file char by char FileReader fr = new FileReader( "New Text Document.txt" ); FileWriter fw = new FileWriter( "AnotherDocumnet.txt" );

int by;

while( ( by = fr.read()) != -1 ){ fw.write( (char) by );

}

fr.close();

fw.close();

}

catch( IOException e ){

System.out.println( e );

}

System.out.println( "Exiting Program..." );

}

}

## OUTPUT

**Source Code:**

**Program 18**

import java.io.File; import java.io.FileReader;

import java.io.BufferedReader; import java.io.IOException;

class LineByLine{

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

File f = new File( "NewDoc.txt" ); FileReader fr = new FileReader( f );

BufferedReader br = new BufferedReader( fr ); String s;

while ( (s = br.readLine()) != null ) {

System.out.println( s );

}

fr.close();

}

catch( IOException e ){ System.out.println( e );

}

System.out.println( "Exiting Program " );

}

}

## OUTPUT

**Source Code:**

**Program 19**

import java.util.Scanner; import java.io.FileWriter; import java.io.IOException;

class writeFile {

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

try{

}

FileWriter fw = new FileWriter( "file.txt" ); fw.write( s );

fw.close();

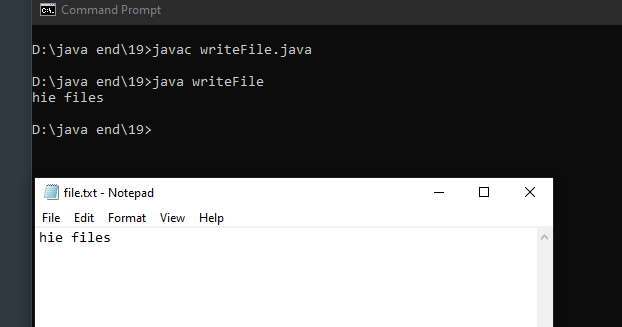
catch( IOException e ){ System.out.println( e );

}

}

}

## OUTPUT



**Source Code:**

**Program 20**

class UserThread extends Thread { int th;

UserThread( int th ){

this.th = th;

}

public void run(){ try{

for( int i = 1; i <= 10; i++ )

System.out.println( "Thread: " + th + "=> " + i );

}

catch( Exception e ){

System.out.println( e );

}

}

}

class MultiThread{

public static void main(String[] args) {

UserThread t1 = new UserThread( 1 ); UserThread t2 = new UserThread( 2 ); UserThread t3 = new UserThread( 3 );

t1.start();

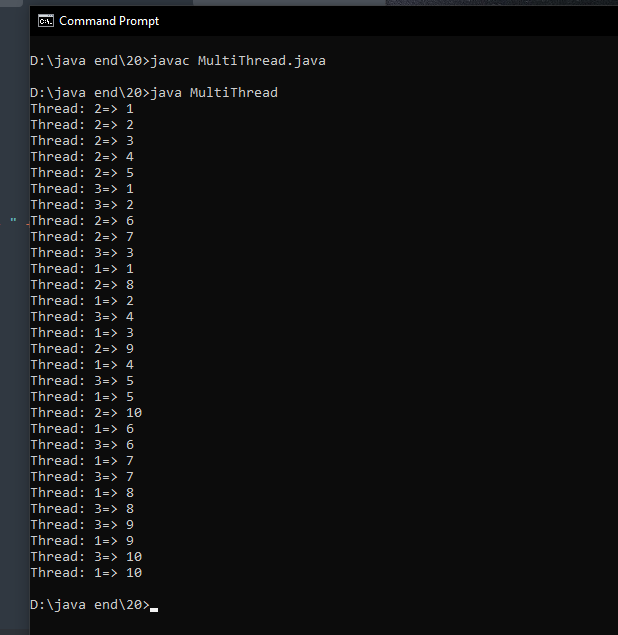
t2.start();

t3.start();

}

}

## OUTPUT



**Source Code:**

**Program 21**

class UserThread extends Thread { public void run(){

System.out.println( "In run" );

}

}

class AboutThread{

public static void main(String[] args) { UserThread t1 = new UserThread(); t1.setName( "Renamed Thread" ); String name = t1.getName(); System.out.println( name );

t1.setPriority( 10 );

int pr = t1.getPriority();

System.out.println( pr );

}

}

## OUTPUT

**Source Code:**

**Program 22**

class UserThread extends Thread{

Display d;

String name;

UserThread( Display d, String name ){ this.d = d;

this.name = name;

}

public void run(){

d.wish( name );

}

}

class Display{

public void wish( String name ){ System.out.print( "\nhello " );

try{

}

Thread.currentThread().sleep( 500 );

catch( Exception e ){

System.out.println( e );

}

System.out.print( name );

}

}

class UsingJoin{

public static void main(String[] args) { Display d = new Display();

UserThread t1 = new UserThread( d, "Nishant" ); UserThread t2 = new UserThread( d, "Aman" );

t1.start();

try{

}

t1.join();

catch( InterruptedException e ){ System.out.println( e );

}

t2.start();

try{

}

t2.join();

catch( InterruptedException e ){ System.out.println( e );

}

}

}

## OUTPUT

**Source Code:**

**Program 23**

class UserThread extends Thread{

Display d;

String name;

UserThread( Display d, String name ){ this.d = d;

this.name = name;

}

public void run(){

d.wish( name );

}

}

class Display{

synchronized void wish( String name ){ System.out.print( "\nhello " );

try{

}

Thread.currentThread().sleep( 1000 );

catch( Exception e ){

System.out.println( e );

}

System.out.print( name );

}

}

class Sync{

public static void main(String[] args) {

UserThread t1 = new UserThread( new Display(), "Nishant" ); UserThread t2 = new UserThread( new Display(), "Aman" );

t1.start(); try{

t1.join();

}

catch( InterruptedException e ){ System.out.println( e );

}

t2.start();

}

}

## OUTPUT

**Source Code:**

**Program 24**

class Thread1 extends Thread{ int amt;

Customer c;

Thread1( Customer c ){ this.amt = 0; this.c = c;

}

public void run(){

c.withdraw( amt );

}

}

class Thread2 extends Thread{ int amt;

Customer c;

Thread2( Customer c ){ this.amt = 0; this.c = c;

}

public void run(){

c.deposit( amt );

}

}

class Customer{

int bal;

synchronized void deposit( int amt ){ bal += amt;

System.out.println( "Updated Balance: " + bal + "; After deposit" );

notify(); // Release waiting thread on current object

}

synchronized void withdraw( int amt ){

try{

be synchronized

}

if( amt > bal )

// Let the other thread finish first

wait(); // Stops current thread, current method should

catch( Exception e ){

System.out.println( e );

}

bal -= amt;

System.out.println( "Updated Balance: " + bal + "; After withdraw" );

}

}

class Comm{

public static void main(String[] args) { Customer c = new Customer();

Thread1 t1 = new Thread1( c ); Thread2 t2 = new Thread2( c );

t1.amt = 1000;

t2.amt = 2000;

t1.start(); // Withdrawl t2.start(); // Deposit

}

}

## OUTPUT

**Program 25**

## Source Code:

import java.awt.\*; import javax.swing.\*;

class B extends JFrame{ B(){

setLayout(new FlowLayout()); JLabel l1 = new JLabel("Name"); JTextField tf1 = new JTextField(20);

JLabel l2 = new JLabel("Section"); JTextField tf2 = new JTextField(20);

JLabel l3 = new JLabel("Univ Roll No"); JTextField tf3 = new JTextField(20);

JLabel l4 = new JLabel("Age"); JTextField tf4 = new JTextField(20);

JLabel l5 = new JLabel("Email ID"); JTextField tf5 = new JTextField(20);

JButton b1 = new JButton("Submit"); add(l1);

add(tf1);

add(l2);

add(tf2);

add(l3);

add(tf3);

add(l4);

add(tf4);

add(l5);

add(tf5);

add(b1);

}

public static void main(String args[]){

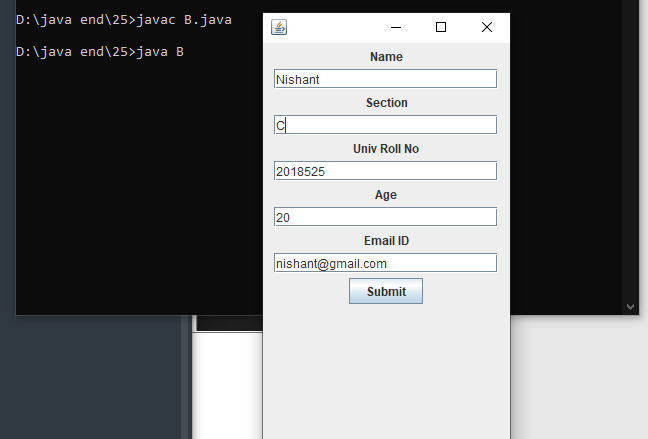
B d = new B();

d.setSize(300,400); d.setVisible(true);

}

}

## OUTPUT



**Program 26**

## Source Code:

import javax.swing.\*; import java.awt.\*;

class AB extends Canvas{

public void paint(Graphics g){ g.drawOval(67,67,50,50); g.setColor(Color.red); g.fillOval(67,67,50,50); g.setColor(Color.black);

g.drawOval(385,67,50,50); g.setColor(Color.red); g.fillOval(385,67,50,50); g.setColor(Color.black);

g.drawOval(50,50,400,400); g.setColor(Color.yellow); g.fillOval(50,50,400,400); g.setColor(Color.black);

g.drawOval(140,140,50,50); g.setColor(Color.white); g.fillOval(140,140,50,50); g.setColor(Color.black);

g.drawOval(300,140,50,50); g.setColor(Color.white); g.fillOval(300,140,50,50); g.setColor(Color.black);

g.drawLine(240,200,240,300); g.drawArc(145,200,200,200,0,-180);

}

public static void main(String args[]){ AB a=new AB();

JFrame f=new JFrame(); f.add(a); f.setSize(400,400); f.setVisible(true);

}

}

## OUTPUT



**Program 27**

## Source Code:

import java.awt.\*; import javax.swing.\*; import java.awt.event.\*;

class B extends JFrame implements ActionListener

{

JTextField tf1, tf2,tf3; B()

{

setLayout(new FlowLayout());

JLabel l1 = new JLabel("Enter No 1"); tf1 = new JTextField(20);

JLabel l2 = new JLabel("Enter No 2"); tf2 = new JTextField(20);

JLabel l3 = new JLabel("Result"); tf3 = new JTextField(20);

JButton b1 = new JButton("Sum"); JButton b2 = new JButton("Sub"); add(l1);

add(tf1);

add(l2);

add(tf2);

add(b1);

add(b2);

add(l3);

add(tf3); b1.addActionListener(this); b2.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

int r= 0;

String s = e.getActionCommand(); if(s.equals("Sum"))

r = ( Integer.parseInt(tf1.getText()) ) + ( Integer.parseInt(tf1.getText()) ); if(s.equals("Sub"))

r = ( Integer.parseInt(tf1.getText()) ) - ( Integer.parseInt(tf1.getText()) ); tf3.setText(Integer.toString(r));

}

public static void main(String args[])

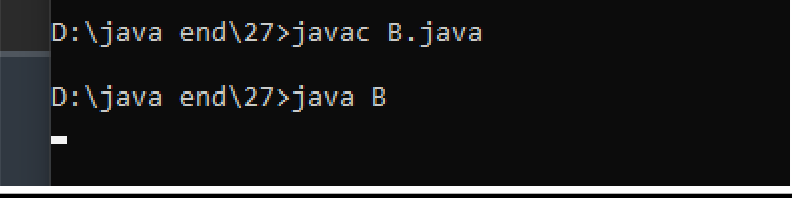
{

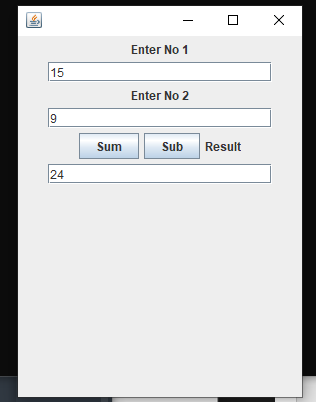
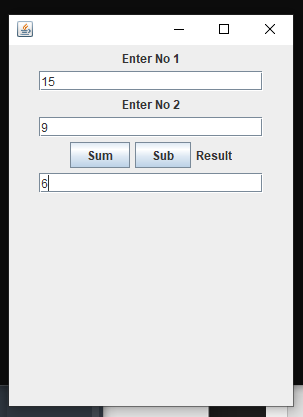
B d = new B(); d.setSize(300,400); d.setVisible(true);

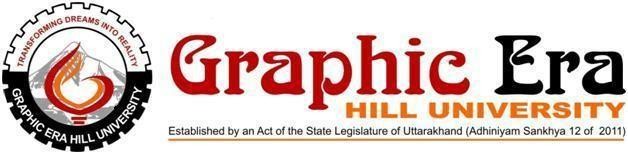
}

}

## OUTPUT







**Term work**

# on

**Java Programming**

# (PCS 408)

**2021-22**

## Submitted to: Submitted by:

Dr. Prateek Srivastava Nishant Bhandari Associate Professor University Roll. No.: GEHU, D.Dun 2018525

Class Roll. No./Section: 35/C

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

## GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.N**  **o.** | **Practical** | **D.O.P.** | **Date of Submiss ion** | **Grade (Viva)** | **Grade (Report File)** | **Total Marks**  **(out of 10)** | **Student’s Signature** | **Teacher’s Signature** |
| **12** | **Practical-12** |  |  |  |  |  |  |  |
| **13** | **Practical-13** |  |  |  |  |  |  |  |
| **14** | **Practical-14** |  |  |  |  |  |  |  |
| **15** | **Practical-15** |  |  |  |  |  |  |  |
| **15** | **Practical-16** |  |  |  |  |  |  |  |
| **17** | **Practical-17** |  |  |  |  |  |  |  |
| **18** | **Practical-18** |  |  |  |  |  |  |  |
| **19** | **Practical-19** |  |  |  |  |  |  |  |
| **20** | **Practical-20** |  |  |  |  |  |  |  |
| **21** | **Practical-21** |  |  |  |  |  |  |  |
| **22** | **Practical-22** |  |  |  |  |  |  |  |
| **23** | **Practical-23** |  |  |  |  |  |  |  |
| **24** | **Practical-24** |  |  |  |  |  |  |  |
| **25** | **Practical-25** |  |  |  |  |  |  |  |
| **26** | **Practical-26** |  |  |  |  |  |  |  |
| **27** | **Practical-27** |  |  |  |  |  |  |  |
| **28** | **Practical-28** |  |  |  |  |  |  |  |