JAGATPUR POST GRADUATE COLLEGE

JAGATPUR VARANASI

SESSION 2021-2022

BCA 3RD YEAR

JAVA PROGRAMMING AND WEB PAGE DESIGN

SUBMITTED BY:- ANKIT PATHAK

Roll no:- 11040206069

SUBMITTED TO :- Dr VIJAYKANT SHARMA

Question: First java program.

```
public class first

{

public static void main(String args[])

{

System.out.println("ankit is the best coder");

}

}
```

Output :-

ankit is the best coder

Question : Write a program to print message “Hello Java!”.

```
class A

{

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

```
{

System.out.println("Hello World!");

}

}
```

Output: -

Hello World!

Question : Write a program with multiple classes.

```
class Room {

float length;

float breadth;

void getdata(float a,float b)

{

length = a;
```

```
breadth = b;

}

}


public class RoomArea {

public static void main(String[] args) {

float area;

Room room1 = new Room(); // Creates an object room1

room1.getdata(14f,10f);    // Assigns values to length and breadth

area = room1.length * room1.breadth;

System.out.println("Area = "+area);

}

}
```

Output: -

```
Area = 140.0
```

Question :Write a program to demonstrate the use of if-else statement.

```
public class IfElseExample {

public static void main(String[] args) {

//defining a variable

int number=13;

//Check if the number is divisible by 2 or not

if(number%2==0){

System.out.println("even number");

}else{

System.out.println("odd number");

}

}

}
```

Output: - odd number

Question : Write a program with multiple statements.

```
import java.lang.Math;

public class SquareRoot {

public static void main(String[] args) {

double x = 5;  // Declaration and Inialization

double y;     // Simple declaration

y = Math.sqrt(x);

System.out.println("y = "+ y);

}

}
```

Output: -

```
y = 2.23606797749979
```

Question :Write a program to demonstrate the use of switch statement.

```
public class SwitchExample {
```

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

//Declaring a variable for switch expression

int number=20;

//Switch expression

switch(number){

//Case statements

case 10: System.out.println("10");

break;

case 20: System.out.println("20");

break;

case 30: System.out.println("30");

break;

//Default case statement

System.out.println("Not in 10, 20 or 30");
```

```
}

}

}
```

Output: -

20

Question :write a program to show swapping and command line argument together in the same program.

```
class swap2

{

public static void main(String args[])

{

int a ,b;

if (args.length==2)
```

```
{

a=Integer.parseInt(args[0]);//first argument converted string to integer

b=Integer.parseInt(args[1]);//second arhguments converted to integer

System.out.println("BEFORE SWAPPING");

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

a=a+b;

b=a-b;

a=a-b;

System.out.println("AFTER SWAPPING");

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

}

else

{

System.out.println("give input as command line argument usage:java swap 12 23");
```

```
return;

}

}

}
```

Output :

```
give input as command line argument usage:java swap 12 23
```

Question: Write a program to demonstrate the use of for loop.

```
public class ForExample {

public static void main(String[] args) {

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

System.out.println(i);

}

}
```

```
}
```

Output: -

```
1

2

3

4

5

6

7

8

9

10
```

Question : Write a program to demonstrate the use of do-while loop.

```
public class DoWhileExample {

public static void main(String[] args) {

int i=1;

do{
        System.out.println(i);
        i++;
        }while(i<=10);
        }
        }
```

Output: -

1

2

3

4

5

6

7

8

9

10

Question : Write a program to demonstrate the break statement.

```
public class BreakExample2 {

public static void main(String[] args) {

for(int i=1;i<=3;i++){

for(int j=1;j<=3;j++){

if(i==2&&j==2){

break;

}

System.out.println(i+" "+j);

}

}

}

}
```

Output: -

1 1

1 2

1 3

2 1

3 1

3 2

3 3

Question : Write a program to demonstrate the use of continue statement.

```
public class ContinueExample {

public static void main(String[] args) {

for(int i=1;i<=3;i++){

for(int j=1;j<=3;j++){

if(i==2&&j==2){

continue;

}

System.out.println(i+" "+j);

}
```

```
}

}

}
```

Output: -

1 1

1 2

1 3

2 1

2 3

3 1

3 2

3 3

Question : Write a program to demonstrate the use of do-while loop.

```
public class CommentExample {

public static void main(String[] args) {

/* Let's declare and
```

```
print variable in java. */

int i=10;

System.out.println(i);

/* float j = 5.9;

float k = 4.4;

System.out.println( j + k ); */

}

}
```

Output: -

10

Question : Write a program to illustrate how to define a class and fields.

```
//Defining a Student class.

class Student{

//defining fields
```

```
int id;//field or data member or instance variable

String name;

//creating main method inside the Student class

public static void main(String args[]){

//Creating an object or instance

Student s1=new Student();//creating an object of Student

//Printing values of the object

System.out.println(s1.id);//accessing member through reference variable

System.out.println(s1.name);

}

}
```

Output: -

0

null

Question : Write a program to initialize through reference.

```
class Student{

int id;

String name;

}

class TestStudent{

public static void main(String args[]){

Student s1=new Student();

s1.id=101;

s1.name="NAKUL";

System.out.println(s1.id+" "+s1.name);//printing members with a white space

}

}
```

Output:-

101 NAKUL

Question : write a program using datainputstream method .

```
import java.io.*;

class test

{

public static void main(String args[])

{

String S=" ";

DataInputStream d=new DataInputStream(System.in);

try

{

System.out.println("Enter your name ");

S=d.readLine();

}

catch(Exception e)

{}

System.out.println(S+ "WELCOME TO THE WORLD OF JAVA");

}
```

```
}
```

Output :

```
Enter your name
anku
ankuWELCOME TO THE WORLD OF JAVA
```

Question :Write a program to initialize through method.

```
class Employee{
int id;
String name;
float salary;
void insert(int i, String n, float s) {
id=i;
name=n;
salary=s;
}
void display(){System.out.println(id+" "+name+" "+salary);
}
```

```
}

public class TestEmployee {

public static void main(String[] args) {

Employee e1=new Employee();

Employee e2=new Employee();

Employee e3=new Employee();

e1.insert(101,"Ajeet",45000);

e2.insert(102,"Irfan",25000);

e3.insert(103,"Nakul",55000);

e1.display();

e2.display();

e3.display();

}

}
```

Output: -

```
101 Ajeet 45000.0

102 Irfan 25000.0
```

103 Nakul 55000.0

Question : write a program using AND operator.

```
class p3

{

public static void main(String args[])

{

int a=10;

int b=5;

int c=20;

System.out.println(a<b&&a++<c);//false && true =false

System.out.println(a);// 10 because seconf condition is not checked

System.out.println(a<b&a++<c);//false && true = false

System.out.println(a);//11 because second condition is checked

}

}
```

Output :

false

10

false

11

Question : Write a program to demonstrate calling a method through anonymous object.

```
class Calculation{

void fact(int  n){

int fact=1;

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

fact=fact*i;

}

System.out.println("factorial is "+fact);

}

public static void main(String args[]){

new Calculation().fact(5);//calling method with anonymous object

}

}
```

Output: -

factorial is 120

Question : Write a program to demonstrate the constructor method.

```
class Bike{

//creating a default constructor

Bike(){System.out.println("Bike is created");}

//main method

public static void main(String args[]){

//calling a default constructor

Bike b=new Bike();

}

}
```

Output: -

Bike is created

Question : Write a program to demonstrate constructor overloading.

```
class Student{

int id;
```

```
String name;

int age;

//creating two arg constructor

Student(int i,String n){

id = i;

name = n;

}

//creating three arg constructor

Student(int i,String n,int a){

id = i;

name = n;

age=a;

}

void display(){System.out.println(id+" "+name+" "+age);}



public static void main(String args[]){

Student s1 = new Student(111,"Karan");

Student s2 = new Student(222,"Aryan",25);
```

```
s1.display();

s2.display();

}

}
```

Output: -

111 Karan 0

222 Aryan 25

Question : Write a program to demonstrate the use of static variable.

```
class Student{

int rollno;//instance variable

String name;

static String college ="ITS";//static variable

//constructor

Student(int r, String n){

rollno = r;

name = n;
```

```
}

//method to display the values

void display (){System.out.println(rollno+" "+name+" "+college);}

}

//Test class to show the values of objects

public class TestStaticVariable{

public static void main(String args[]){

Student s1 = new Student(111,"Karan");

Student s2 = new Student(222,"Aryan");



s1.display();

s2.display();

}

}
```

Output: -

111 Karan ITS

222 Aryan ITS

Question : Write a program to demonstrate the use of an instance variable which gets memory each time when we create an object of the class.

```
class Counter{

int count=0;//will get memory each time when the instance is created



Counter(){

count++;//incrementing value

System.out.println(count);

}

public static void main(String args[]){

//Creating objects

Counter c1=new Counter();

Counter c2=new Counter();

Counter c3=new Counter();

}

}
```

Output: -

1

1

1

Question : Write a program to demonstrate the use this keyword.

```
class Student{
int rollno;
String name;
float fee;
Student(int rollno,String name,float fee){
this.rollno=rollno;
this.name=name;
this.fee=fee;
}
void display(){System.out.println(rollno+" "+name+" "+fee);}
}

class TestThisKeyword{
```

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

Student s1=new Student(111,"Ankit",5000f);

Student s2=new Student(112,"Sumit",6000f);

s1.display();

s2.display();

}

}
```

Output: -

111 Ankit 5000.0

112 Sumit 6000.0

Question : Write a program to illustrate inheritance.

```
class Employee{

float salary=40000;

}

class Programmer extends Employee{

int bonus=10000;
```

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

Programmer p=new Programmer();

System.out.println("Programmer salary is:"+p.salary);

System.out.println("Bonus of Programmer is:"+p.bonus);

}

}
```

Output: -

```
Programmer salary is:40000.0

Bonus of Programmer is:10000
```

Question :Write a program to illustrate multi-level inheritance.

```
class Animal{

void eat(){System.out.println("eating...");

}

}

class Dog extends Animal{

void bark(){System.out.println("barking...");
```

```
}

}

class BabyDog extends Dog{

void weep(){System.out.println("weeping...");

}

}

class TestInheritance{

public static void main(String args[]){

BabyDog d=new BabyDog();

d.weep();

d.bark();

d.eat();

}

}
```

Output: -

```
weeping...

barking...
```

eating...

Question : write programme to print Fibonacci series.

```
class fibonacci

{

static int n1=0,n2=1,n3=0;

static void printfibonacci(int count)

{

if (count>0)

{

n3=n1+n2;

n1=n2;

n2=n3;

System.out.print(" "+n3);

printfibonacci(count-1);

}

}

public static void main(String srgs[])

{
```

```
int count=10;

System.out.println(n1+" "+n2);// printing 0 and 1

printfibonacci(count-2);//n-2 because 2 numbers are already printed

}

}
```

Output : 0 1

1 2 3 5 8 13 21 34

Question : Write a program to illustrate Hierarchical Inheritance.

```
class Animal{

void eat(){System.out.println("eating...");

}

}

class Dog extends Animal{

void bark(){System.out.println("barking...");

}

}
```

```
class Cat extends Animal{

void meow(){System.out.println("meowing...");

}

}

class TestInheritance{

public static void main(String args[]){

Cat c=new Cat();

c.meow();

c.eat();

}

}
```

Output: -

```
meowing...

eating...
```

Question : Write a program to demonstrate the concept of aggregation.

```
class Operation{
```

```
int square(int n){
return n*n;
}
}

class Circle{
Operation op;  //aggregation
double pi=3.14;

double area(int radius){
op=new Operation();
int rsquare=op.square(radius);  //code reusability
return pi*rsquare;
}

public static void main(String args[]){
Circle c=new Circle();
double result=c.area(5);
```

```
System.out.println(result);
}
}
```

Output: -

78.5

Question :write a program to swap the numbers.

```
class swap
/*swap two numbers without using any temp variables*/
{
public static void main(String args[])
{
int a,b;
a=1;
b=2;
System.out.println("Before swapping [exchange]");
System.out.println("a="+a+"b="+b);
a=a+b;
```

```
b=a-b;

a=a-b;

System.out.println("After swapping [exchange]");

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

}}
```

Output :

```
Before swapping [exchange]

a=1b=2

After swapping [exchange]

a=2b=1
```

Question : Write a program to demonstrate the concept of aggregation.

```
class Overloading{

void sum(int a,long b){System.out.println(a+b);

}
```

```
void sum(int a,int b,int c){System.out.println(a+b+c);

}



public static void main(String args[]){

Overloading obj=new Overloading();

obj.sum(20,20);

obj.sum(20,20,20);

}

}
```

Output: -

40

60

Question : Write a program to demonstrate the concept of method overriding.

```
class Vehicle{

//defining a method

void run(){System.out.println("Vehicle is running");

}

}
```

```
//Creating a child class

class Bike extends Vehicle{

//defining the same method as in the parent class

void run(){System.out.println("Bike is running safely"

}



public static void main(String args[]){

Bike obj = new Bike();//creating object

obj.run();//calling method

}

}
```

Output: -

Bike is running safely

Question : write a program using IOException.

```
import java.io.*;

public class five

{
```

```
public static void main(String args[])throws IOException

{

InputStreamReader ir=new InputStreamReader(System.in);

BufferedReader br=new BufferedReader(ir);

System.out.println("enter your name master killer");

String name=br.readLine();

System.out.println("Hello the legendary killer "+ name +" !");

}

}
```

Output :

```
enter your name master killer

ankit

Hello the legendary killer ankit !
```

Question : Write a program to demonstrate the use of super keyword.

```
class Animal{

void eat(){System.out.println("eating...");
```

```
}

}

class Dog extends Animal{

void eat(){System.out.println("eating bread...");

}

void bark(){System.out.println("barking...");

}

void work(){

super.eat();

bark();

}

}

class TestSuper{

public static void main(String args[]){

Dog d=new Dog();

d.work();

}

}
```

Output: -

eating...

barking...

Question : Write a program to demonstrate the use of final keyword.

```
class Bike{

final int speedlimit=90; //final variable

void run(){

speedlimit=400;

}

public static void main(String args[]){

Bike obj=new  Bike();

obj.run();

}

}
```

Output: -

Bike.java:4: error: cannot assign a value to final variable speedlimit

speedlimit=400;

^

1 error

Question : write a program using shift operator.

```
class p4
{
public static void main(String args[])
{
System.out.println(10<<2);// 10*2^2=10*4=40
System.out.println(10<<3);//10*2^3=10*8=80
System.out.println(20<<2);//20*2^2=20*4=80
}
}
```

Output :

40

80

80

Question :Write a program to demonstrate the concept of runtime polymorphism.

```
class Shape{

void draw(){System.out.println("drawing...");}

}

class Rectangle extends Shape{

void draw(){System.out.println("drawing rectangle...");

}

}

class Circle extends Shape{

void draw(){System.out.println("drawing circle...");

}

}

class Triangle extends Shape{

void draw()

{
```

```
System.out.println("drawing triangle...");

}

}

class TestPolymorphism{

public static void main(String args[]){

Shape s;

s=new Rectangle();

s.draw();

s=new Circle();

s.draw();

s=new Triangle();

s.draw();

}

}
```

Output: -

drawing rectangle...

drawing circle... drawing triangle...

Question : Write a program to demonstrate dynamic binding.

```
class Animal{

void eat(){

System.out.println("animal is eating...");

}

}


class Dog extends Animal{

void eat(){

System.out.println("dog is eating...");

}


public static void main(String args[]){

Animal a=new Dog();

a.eat();

}
```

}

Output: -

dog is eating...

Question : Write a program to demonstrate instance of term.

class Animal{}

class Dog extends Animal{//Dog inherits Animal

public static void main(String args[]){

Dog d=new Dog();

System.out.println(d instanceof Animal);//true

}

}

Output: -

true

Question :write a programme to implement all the arithmatical operations

```
class arithmatic

{

public static void main(String args[])

{

int x=100,y=20;

float a=12.5f,b=3.0f;

System.out.println("x="+x);

System.out.println("y="+y);

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

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

System.out.println("x+y="+(x+y));

System.out.println("x-y="+(x-y));

System.out.println("x*y="+(x*y));

System.out.println("x/y="+(x/y));

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

System.out.println("a-b="+(a-b));
```

*System.out.println("a*b="+(a*b));*

System.out.println("a/b="+(a/b));

System.out.println("a%b="+(a%b));

System.out.println("all your current arithmatical operaton are there calculated because you have given the arguments at compile time");

}

}

Output:

x=100

y=20

a=12.5

b=3.0

x+y=120

x-y=80

*x*y=2000*

x/y=5

a+b=15.5

a-b=9.5

*a*b=37.5*

a/b=4.1666665

a%b=0.5

all your current arithmatical operaton are there calculated because you have given the arguments at compile time

Question : Write a program using abstract class.

```
abstract class Bike{

Bike(){System.out.println("bike is created");}

abstract void run();

void changeGear(){System.out.println("gear changed");

}

}

//Creating a Child class which inherits Abstract class

class Honda extends Bike{

void run(){System.out.println("running safely..");

}

}

//Creating a Test class which calls abstract and non-abstract methods

class TestAbstraction{
```

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

Bike obj = new Honda();

obj.run();

obj.changeGear();

}

}
```

Output: -

```
bike is created

running safely..

gear change...
```

Question :Write a program to demonstrate interface.

```
interface Drawable{

void draw();

}

//Implementation: by second user
```

```
class Rectangle implements Drawable{

public void draw(){System.out.println("drawing rectangle");

}

}

class Circle implements Drawable{

public void draw(){System.out.println("drawing circle");

}

}

//Using interface: by third user

class TestInterface{

public static void main(String args[]){

Drawable d=new Circle();//In real scenario, object is provided by method e.g. getDrawable()

d.draw();

}

}
```

Output: -

```
drawing circle
```

Question :Write a program to demonstrate abstract class with interface.

```
interface A{

void a();//bydefault, public and abstract

void b();

void c();

void d();

}



//Creating abstract class that provides the implementation of one method of A interface

abstract class B implements A{

public void c(){System.out.println("I am C");

}

}



//Creating subclass of abstract class, now we need to provide the implementation of rest of the methods

class M extends B{

public void a(){System.out.println("I am a");
```

```
}

public void b(){System.out.println("I am b");

}

public void d(){System.out.println("I am d");

}

}


//Creating a test class that calls the methods of A interface

class Test{

public static void main(String args[]){

A a=new M();

a.a();

a.b();

a.c();

a.d();

}

}
```

Output: -

I am a

I am b

I am C

I am d

Question :Write a program to demonstrate the use of package.

//save by A.java

package pack;

public class A{

public void msg(){System.out.println("Hello");

}

}

//save by B.java

package mypack;

import pack.;*

class B{

public static void main(String args[]){

A obj = new A();

obj.msg();

}

}

Output: -

Hello

Question : Write a program to demonstrate access modifier.

//save by A.java

package pack;

class A{

void msg(){System.out.println("Hello");

}

}

```
//save by B.java

package mypack;

import pack.*;

class B{

public static void main(String args[]){

A obj = new A();//Compile Time Error

obj.msg();//Compile Time Error

}

}
```

Output: -

```
Compilation Error!
```

Question :Write a program to demonstrate encapsulation.

```
//A Account class which is a fully encapsulated class.

//It has a private data member and getter and setter methods.
```

```
class Account {

//private data members

private long acc_no;

private String name,email;

private float amount;

//public getter and setter methods

public long getAcc_no() {

return acc_no;

}

public void setAcc_no(long acc_no) {

this.acc_no = acc_no;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {
```

```
return email;

}

public void setEmail(String email) {

this.email = email;

}

public float getAmount() {

return amount;

}

public void setAmount(float amount) {

this.amount = amount;

}

}
```

```
//A Java class to test the encapsulated class Account.

public class TestEncapsulation {

public static void main(String[] args) {

//creating instance of Account class
```

```
Account acc=new Account();

//setting values through setter methods

acc.setAcc_no(7560504000L);

acc.setName("Sonoo Jaiswal");

acc.setEmail("sonoojaiswal@gmail.com");

acc.setAmount(500000f);

//getting values through getter methods

System.out.println(acc.getAcc_no()+" "+acc.getName()+" "+acc.getEmail()+"
"+acc.getAmount());

}

}
```

Output: -

```
7560504000 Sonoo Jaiswal sonoojaiswal@gmail.com 500000.0
```

Question : Write a program to implement multi-dimensional array.

```
class Testarray{

public static void main(String args[]){

//declaring and initializing 2D array

int arr[][]={{1,2,3},{2,4,5},{4,4,5}};

//printing 2D array

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

System.out.print(arr[i][j]+" ");

}

System.out.println();

}

}

}
```

Output: -

```
1 2 3
```

2 4 5

4 4 5

Question :Write a program to demonstrate the use of Math class.

```
public class JavaMathExample

{

public static void main(String[] args)

{

double x = 28;

double y = 4;


// return the maximum of two numbers

System.out.println("Maximum number of x and y is: " +Math.max(x, y));


// return the square root of y

System.out.println("Square root of y is: " + Math.sqrt(y));


//returns 28 power of 4 i.e. 28*28*28*28

System.out.println("Power of x and y is: " + Math.pow(x, y));
```

```
// return the logarithm of given value

System.out.println("Logarithm of x is: " + Math.log(x));

System.out.println("Logarithm of y is: " + Math.log(y));


// return the logarithm of given value when base is 10

System.out.println("log10 of x is: " + Math.log10(x));

System.out.println("log10 of y is: " + Math.log10(y));


// return the log of x + 1

System.out.println("log1p of x is: " +Math.log1p(x));


// return a power of 2

System.out.println("exp of a is: " +Math.exp(x));


// return (a power of 2)-1

System.out.println("expm1 of a is: " +Math.expm1(x));

}

}
```

Output: -

Maximum number of x and y is: 28.0

Square root of y is: 2.0

Power of x and y is: 614656.0

Logarithm of x is: 3.332204510175204

Logarithm of y is: 1.3862943611198906

log10 of x is: 1.4471580313422192

log10 of y is: 0.6020599913279624

log1p of x is: 3.367295829986474

exp of a is: 1.446257064291475E12

expm1 of a is: 1.446257064290475E12

Question : Write a program to demonstrate the use of Wrapper classes.

```
public class WrapperExample{

public static void main(String args[]){

byte b=10;

short s=20;

int i=30;
```

```
long l=40;

float f=50.0F;

double d=60.0D;

char c='a';

boolean b2=true;



//Autoboxing: Converting primitives into objects

Byte byteobj=b;

Short shortobj=s;

Integer intobj=i;

Long longobj=l;

Float floatobj=f;

Double doubleobj=d;

Character charobj=c;

Boolean boolobj=b2;



//Printing objects

System.out.println("---Printing object values---");

System.out.println("Byte object: "+byteobj);
```

```
System.out.println("Short object: "+shortobj);

System.out.println("Integer object: "+intobj);

System.out.println("Long object: "+longobj);

System.out.println("Float object: "+floatobj);

System.out.println("Double object: "+doubleobj);

System.out.println("Character object: "+charobj);

System.out.println("Boolean object: "+boolobj);


//Unboxing: Converting Objects to Primitives

byte bytevalue=byteobj;

short shortvalue=shortobj;

int intvalue=intobj;

long longvalue=longobj;

float floatvalue=floatobj;

double doublevalue=doubleobj;

char charvalue=charobj;

boolean boolvalue=boolobj;


//Printing primitives
```

```
System.out.println("---Printing primitive values---");

System.out.println("byte value: "+bytevalue);

System.out.println("short value: "+shortvalue);

System.out.println("int value: "+intvalue);

System.out.println("long value: "+longvalue);

System.out.println("float value: "+floatvalue);

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

System.out.println("char value: "+charvalue);

System.out.println("boolean value: "+boolvalue);

}

}
```

Output: -

```
---Printing object values---

Byte object: 10

Short object: 20

Integer object: 30

Long object: 40
```

Float object: 50.0

Double object: 60.0

Character object: a

Boolean object: true

---Printing primitive values---

byte value: 10

short value: 20

int value: 30

long value: 40

float value: 50.0

double value: 60.0

char value: a

boolean value: true

Question : Write a program to Demonstrate Recursion.

```
public class RecursionExample {

static int factorial(int n){

if (n == 1)

return 1;
```

```
else

return(n * factorial(n-1));

}


public static void main(String[] args) {

System.out.println("Factorial of 5 is: "+factorial(5));

}

}
```

Output: -

```
Factorial of 5 is: 120
```

Question :Write a program to print Fibonacci Series.

```
public class Fibonacci {

static int n1=0,n2=1,n3=0;

static void printFibo(int count){

if(count>0){

n3 = n1 + n2;
```

```
n1 = n2;

n2 = n3;

System.out.print(" "+n3);

printFibo(count-1);

}

}


public static void main(String[] args) {

int count=15;

System.out.print(n1+" "+n2);//printing 0 and 1

printFibo(count-2);//n-2 because 2 numbers are already printed

}

}
```

Output: -

```
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

Question :Write a program to implement call by value.

```
class Operation{

int data=50;


void change(int data){

data=data+100;//changes will be in the local variable only

}


public static void main(String args[]){

Operation op=new Operation();


System.out.println("before change "+op.data);

op.change(500);

System.out.println("after change "+op.data);

}

}
```

Output: -

before change 50

after change 50

Question :Write a program to implement call by reference.

```
class Operation{

int data=50;


void change(Operation op){

op.data=op.data+100;//changes will be in the instance variable

}


public static void main(String args[]){

Operation op=new Operation();


System.out.println("before change "+op.data);

op.change(op);//passing object

System.out.println("after change "+op.data);

}
```

}

Output: -

before change 50

after change 150

Question : Write a program to demonstrate Command Line Arguments.

class A{

public static void main(String args[]){

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

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

}

}

Output: -

ANKIT pathak

1

3

abc

Question : Write a program to demonstrate the use of String Datatype.

```
public class StringExample{

public static void main(String args[]){

String s1="java";//creating string by Java string literal

char ch[]={'s','t','r','i','n','g','s'};

String s2=new String(ch);//converting char array to string

String s3=new String("example");//creating Java string by new keyword

System.out.println(s1);

System.out.println(s2);

System.out.println(s3);

}
```

}

Output: -

java

strings

example

Question :Write a program to demonstrate that strings are immutable.

class Testimmutablestring{

public static void main(String args[]){

String s="Sachin";

s.concat(" Tendulkar");//concat() method appends the string at the end

System.out.println(s);//will print Sachin because strings are immutable objects

}

}

Output: -

Sachin

Question :Write a program to demonstrate string comparison.

```
class Teststringcomparison{

public static void main(String args[]){

String s1="Sachin";

String s2="Sachin";

String s3=new String("Pawan");

String s4="Saurav";

System.out.println(s1.equals(s2));//true

System.out.println(s1.equals(s3));//false

System.out.println(s1.equals(s4));//false

}

}
```

Output: -

true

false

false

Question : Write a program to demonstrate concatenation in string.

```
public class ConcatExample{

public static void main(String args[]){

String s1="java string";

// The string s1 does not get changed, even though it is invoking the method

// concat(), as it is immutable. Therefore, the explicit assignment is required here.

s1.concat("is immutable");

System.out.println(s1);

s1=s1.concat(" is immutable so assign it explicitly");

System.out.println(s1);

}

}
```

Output: -

java string

java string is immutable so assign it explicitly

Question : Write a program to check whether a string is empty or not .

```
public class IsEmptyExample{

public static void main(String args[]){

String s1="";

String s2="javatpoint";


System.out.println(s1.isEmpty());

System.out.println(s2.isEmpty());

}

}
```

Output: -

```
true

false
```

Question :Write a program to find out the length of any string .

```
public class LengthExample{

public static void main(String args[]){

String s1="javatpoint";

String s2="python";

System.out.println("string length is: "+s1.length());//10 is the length of javatpoint string

System.out.println("string length is: "+s2.length());//6 is the length of python string

}

}
```

Output: -

```
string length is: 10

string length is: 6
```

Question : Write a program to convert string into UpperCase and LowerCase.

```
public class Main {

public static void main(String[] args) {

String txt = "Hello World";
```

```
System.out.println(txt.toUpperCase());

System.out.println(txt.toLowerCase());

}

}
```

Output: -

```
HELLO WORLD

hello world
```

Question :Write a program to demonstrate exception handling.

```
public class JavaExceptionExample{

public static void main(String args[]){

try{

int data=100/0;

}

catch(ArithmeticException e)

{

System.out.println(e);
```

}

System.out.println("rest of the code...");

}

}

Output: -

java.lang.ArithmeticException: / by zero

rest of the code...

Question :Write a program to implement try-catch block.

public class TryCatchExample {

public static void main(String[] args) {

try

{

int arr[]= {1,3,5,7};

System.out.println(arr[10]); //may throw exception

}

// handling the array exception

```
catch(ArrayIndexOutOfBoundsException e)

{

System.out.println(e);

}

System.out.println("rest of the code");

}

}
```

Output: -

```
java.lang.ArrayIndexOutOfBoundsException: Index 10 out of bounds for length 4

rest of the code
```

Question :Write a program to implement try-catch block.

```
public class MultipleCatchBlock {

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

```
try{
int a[]=new int[5];
a[5]=30/0;
}
catch(ArithmeticException e)
{
System.out.println("Arithmetic Exception occurs");
}
catch(ArrayIndexOutOfBoundsException e)
{
System.out.println("ArrayIndexOutOfBounds Exception occurs");
}
catch(Exception e)
{
System.out.println("Parent Exception occurs");
}
System.out.println("rest of the code");
}
}
```

Output: -

Arithmetic Exception occurs

rest of the code

Question :Write a program to implement Nested Try Block.

```
public class NestedTryBlock{

public static void main(String args[]){

//outer try block

try{

//inner try block 1

try{

System.out.println("going to divide by 0");

int b =39/0;

}

//catch block of inner try block 1

catch(ArithmeticException e)

{
```

```
System.out.println(e);

}

//inner try block 2

try{

int a[]=new int[5];

//assigning the value out of array bounds

a[5]=4;

}

//catch block of inner try block 2

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println(e);

}

System.out.println("other statement");

}

//catch block of outer try block

catch(Exception e)

{

System.out.println("handled the exception (outer catch)");
```

}

System.out.println("normal flow..");

}

}

Output: -

going to divide by 0

java.lang.ArithmeticException: / by zero

java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5

other statement

normal flow..

Question : Write a program to implement finally block.

public class TestFinallyBlock{

public static void main(String args[]){

try {

```
System.out.println("Inside the try block");

//below code throws divide by zero exception

int data=25/0;

System.out.println(data);

}

//cannot handle Arithmetic type exception

//can only accept Null Pointer type exception

catch(NullPointerException e){

System.out.println(e);

}

//executes regardless of exception occured or not

finally {

System.out.println("finally block is always executed");

}

System.out.println("rest of the code...");

}

}
```

Output: -

Inside the try block

finally block is always executed

Exception in thread "main" java.lang.ArithmeticException: / by zero

at TestFinallyBlock.main(TestFinallyBlock.java:8)

Question : Write a program to implement throw keyword.

```
public class TestThrow {
//function to check if person is eligible to vote or not
public static void validate(int age) {
if(age<18) {
//throw Arithmetic exception if not eligible to vote
throw new ArithmeticException("Person is not eligible to vote");
}
else {
System.out.println("Person is eligible to vote!!");
}
}
```

```
//main method

public static void main(String args[]){

//calling the function

validate(13);

System.out.println("rest of the code...");

}

}
```

Output: -

```
Exception in thread "main" java.lang.ArithmeticException: Person is not eligible to vote

at TestThrow.validate(TestThrow.java:6)

at TestThrow.main(TestThrow.java:15)
```

Question :Write a program to implement throws keyword.

```
import java.io.*;

class M{

void method()throws IOException{

System.out.println("device operation performed");
```

```
}

}


class TestThrows{

public static void main(String args[])throws IOException{//declare exception

M m=new M();

m.method();

System.out.println("normal flow...");

}

}
```

Output: -

```
device operation performed

normal flow...
```

Question : Write a program to implement LocalDate Class.

```
import java.time.LocalDate;

public class LocalDateExample {
```

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

LocalDate date = LocalDate.now();

LocalDate yesterday = date.minusDays(1);

LocalDate tomorrow = yesterday.plusDays(2);

System.out.println("Today date: "+date);

System.out.println("Yesterday date: "+yesterday);

System.out.println("Tomorrow date: "+tomorrow);

}

}
```

Output: -

```
Today date: 2021-11-14

Yesterday date: 2021-11-13

Tomorrow date: 2021-11-15
```

Question : Write a program to implement LocalTime Class.

```
import java.time.LocalTime;

public class LocalTimeExample {
```

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

LocalTime time = LocalTime.now();

System.out.println(time);

}

}
```

Output: -

```
23:47:04.61525090
```

Q61. Write a program to implement LocalDateTime Class.

```
import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

public class LocalDateTimeExample {

public static void main(String[] args) {

LocalDateTime now = LocalDateTime.now();

System.out.println("Before Formatting: " + now);

DateTimeFormatter format = DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss");

String formatDateTime = now.format(format);
```

```
System.out.println("After Formatting: " + formatDateTime);

}

}
```

Output: -

```
PS C:\Users\Deepak Singh\Desktop\Project> java LocalDateTimeExample

Before Formatting: 2021-11-14T23:50:36.707306300

After Formatting: 14-11-2021 23:50:36
```

Q62. Write a program to demonstrate the use of Date Constructor.

```
import java.util.Date;

public class DateExample {


public static void main(String args[]) {

// Instantiate a Date object

Date date = new Date();



// display time and date using toString()
```

```
System.out.println(date.toString());

}

}
```

Output: -

```
PS C:\Users\Deepak Singh\Desktop\Project> java DateExample

Mon Nov 15 18:20:15 IST 2021
```

Q63. Write a program to demonstrate the use of ArrayList class.

```
import java.util.ArrayList;

import java.util.Collections;  // Import the Collections class
```

```
public class ArrayListExample {

public static void main(String[] args) {

ArrayList<Integer> myNumbers = new ArrayList<Integer>();

myNumbers.add(33);

myNumbers.add(15);

myNumbers.add(20);

myNumbers.add(34);

myNumbers.add(8);

myNumbers.add(12);


Collections.sort(myNumbers);  // Sort myNumbers


for (int element : myNumbers) {

System.out.println(element);

}

}

}
```

Output: -

```
PS C:\Users\Deepak Singh\Desktop\Project> java ArrayListExample

8

12

15

20

33

34
```

Q64. Write a program to implement the concept of thread by extending Thread Class.

```
public class ThreadExample extends Thread {

public static void main(String[] args) {

ThreadExample thread = new ThreadExample();

thread.start();

System.out.println("This code is outside of the thread");

}

public void run() {

System.out.println("This code is running in a thread");

}

}
```

Output: -

```
PS C:\Users\Deepak Singh\Desktop\Project> java ThreadExample

This code is outside of the thread

This code is running in a thread
```

Q65. Write a program to implement the concept of thread by implementing Runnable interface.

```
public class ThreadExample implements Runnable {

public static void main(String[] args) {

ThreadExample obj = new ThreadExample();

Thread thread = new Thread(obj);

thread.start();

System.out.println("This code is outside of the thread");

}

public void run() {

System.out.println("This code is running in a thread");

}

}
```

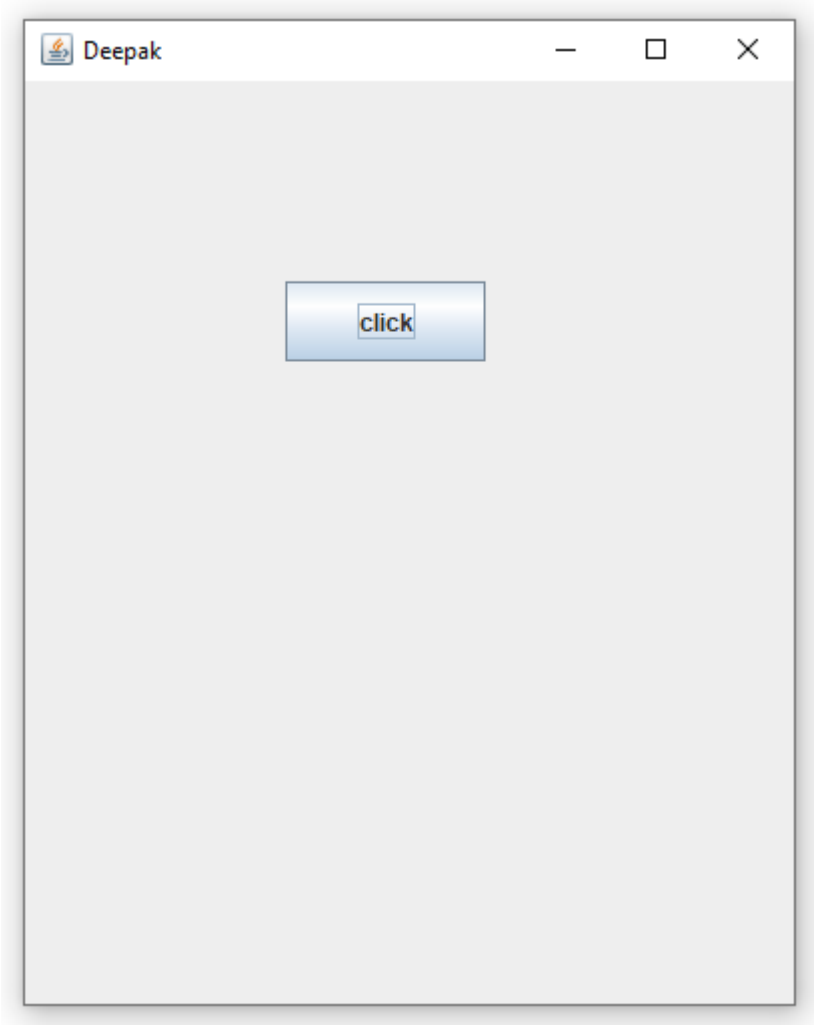
Output: -

```
PS C:\Users\Deepak Singh\Desktop\Project> java ThreadExample

This code is outside of the thread

This code is running in a thread
```

Q66. Write a program show an example of JFrame.

```
import javax.swing.*;

public class FirstSwingExample {

public static void main(String[] args) {

JFrame f=new JFrame();//creating instance of JFrame

JButton b=new JButton("click");//creating instance of JButton

b.setBounds(130,100,100, 40);//x axis, y axis, width, height

f.add(b);//adding button in JFrame

f.setTitle("Deepak");//Setting the Title of the frame

f.setSize(400,500);//400 width and 500 height

f.setLayout(null);//using no layout managers

f.setVisible(true);//making the frame visible

}

}
```

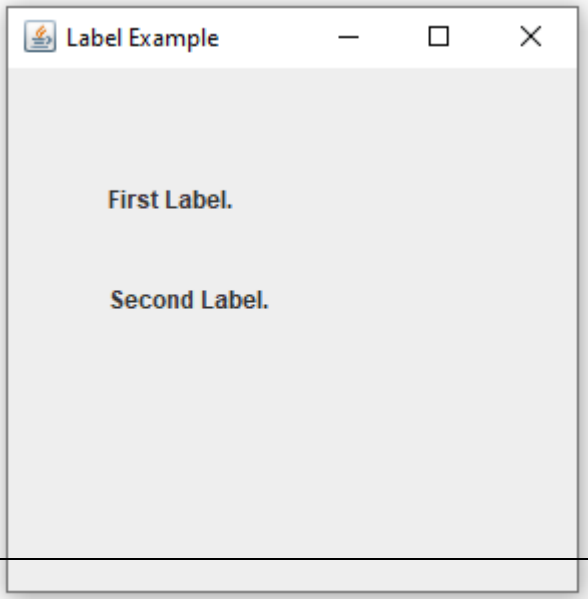
Output: -



Q67. Write a program show an example of JLabel.

```
import javax.swing.*;

class LabelExample

{

public static void main(String args[])

{

JFrame f= new JFrame("Label Example");

JLabel l1,l2;

l1=new JLabel("First Label.");

l1.setBounds(50,50, 100,30);

l2=new JLabel("Second Label.");

l2.setBounds(50,100, 100,30);

f.add(l1); f.add(l2);

f.setSize(300,300);

f.setLayout(null);

f.setVisible(true);

}

}
```

Output: -

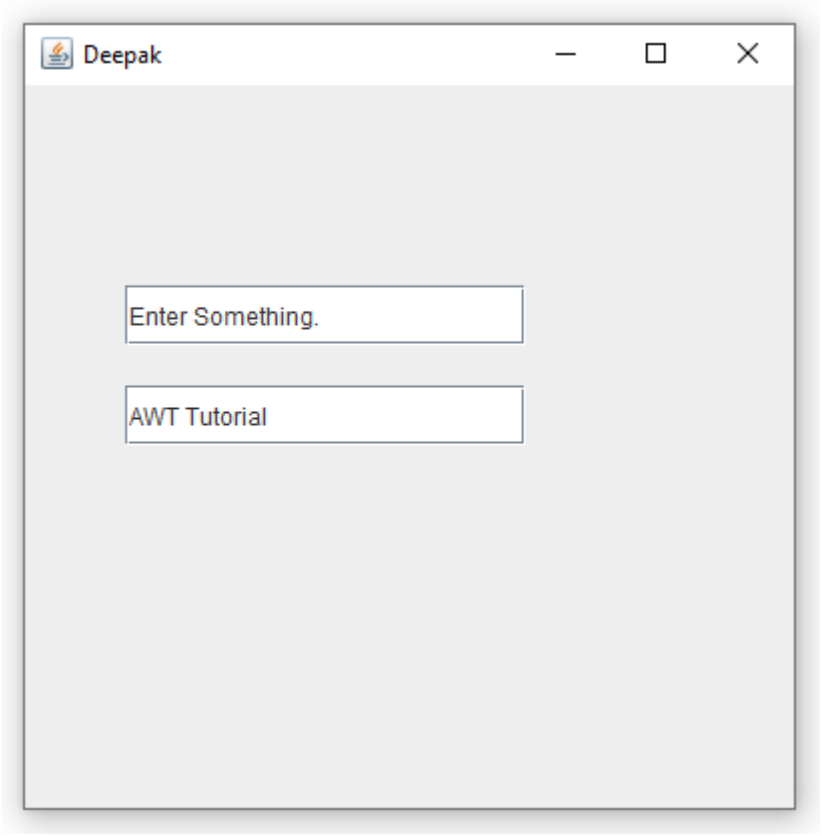


Q68. Write a program show an example of JLabel.

```
import javax.swing.*;

class TextFieldExample

{
```

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

{

JFrame f= new JFrame("TextField Example");

JTextField t1,t2;

t1=new JTextField("Enter Something.");

t1.setBounds(50,100, 200,30);

t2=new JTextField("AWT Tutorial");

t2.setBounds(50,150, 200,30);

f.setTitle("Deepak");

f.add(t1); f.add(t2);

f.setSize(400,400);

f.setLayout(null);

f.setVisible(true);

}

}
```

Output: -

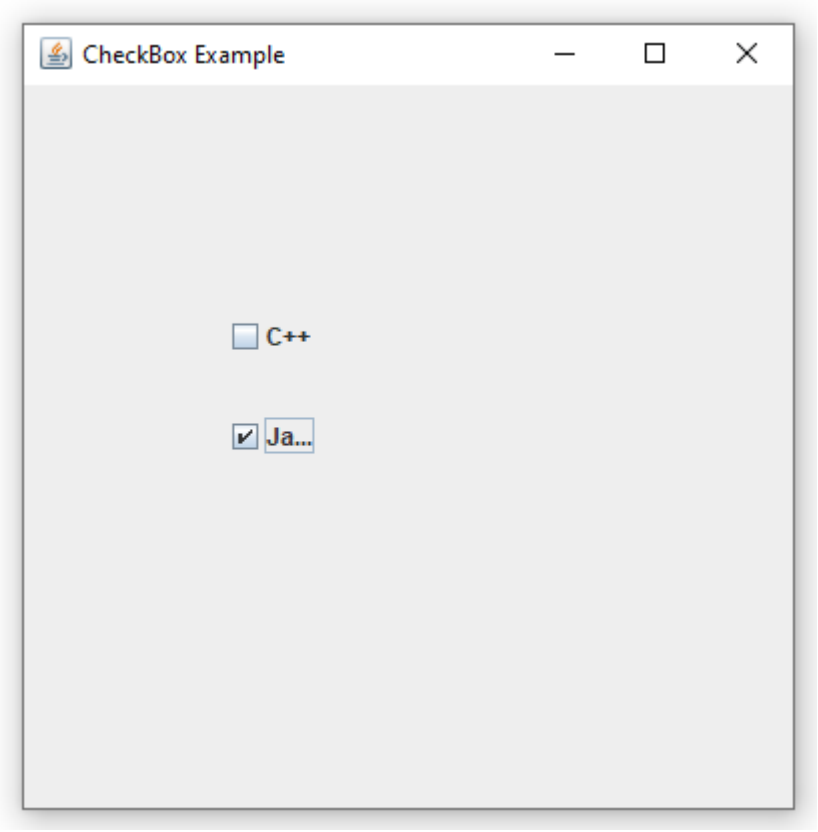


Q69. Write a program show an example of Checkbox.

```
import javax.swing.*;

public class CheckBoxExample

{
```

```
CheckBoxExample(){

JFrame f= new JFrame("CheckBox Example");

JCheckBox checkBox1 = new JCheckBox("C++",false);

checkBox1.setBounds(100,100, 50,50);

JCheckBox checkBox2 = new JCheckBox("Java", true);

checkBox2.setBounds(100,150, 50,50);

f.add(checkBox1);

f.add(checkBox2);

f.setSize(400,400);

f.setLayout(null);

f.setVisible(true);

}

public static void main(String args[])

{

new CheckBoxExample();

}

}
```

Output: -

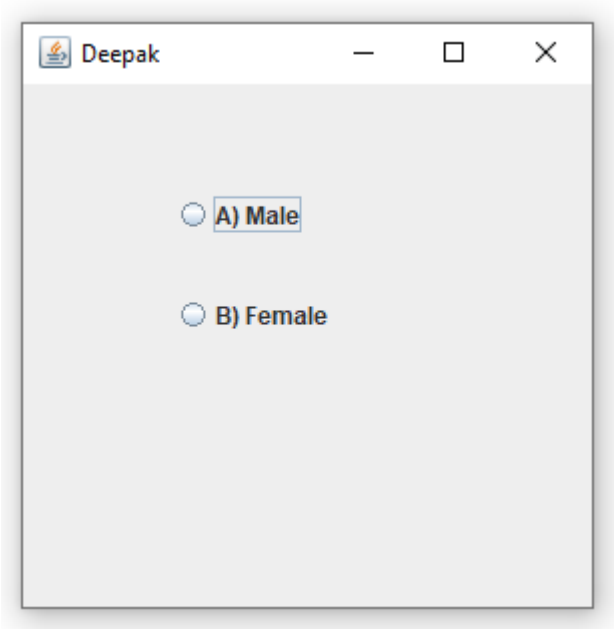


Q70. Write a program show an example of JLabel.

```
import javax.swing.*;

public class RadioButtonExample {

JFrame f;

RadioButtonExample(){
```

```
f=new JFrame();

JRadioButton r1=new JRadioButton("A) Male");

JRadioButton r2=new JRadioButton("B) Female");

r1.setBounds(75,50,100,30);

r2.setBounds(75,100,100,30);

ButtonGroup bg=new ButtonGroup();

bg.add(r1);bg.add(r2);

f.add(r1);f.add(r2);

f.setSize(300,300);

f.setLayout(null);

f.setVisible(true);

}

public static void main(String[] args) {

new RadioButtonExample();

}

}
```

Output: -


Deepak
A) Male
B) Female