21) WAP to Add two number using class.

⇒

```
import java.io.*;

class Add
{
int x, y, z;
}

class Demo
{
public static void main(String args[])
{
Add A1 = new Add();

a1.x = 10;
a1.y = 20;

a1.z = a1.x + a1.y;

System.out.println("Sum =" + a1.z);
}
}
```

Shot on OnePlus
By Yash Bule

22) WAP to Addition of two number by using Method

⇒
```
import java.io.*;

class Add
{
int x, y, z;

void init()
{
x=10;
y=20;
}
void Sum()
{
z=x+y
}
void Display()
{
System.out.println("Addition = " +z);
}
}
class Demo
{
public static void main(String args[])
{
Add a1 = new Add();

a1.init();
a1.Sum();
a1.Display();
}}
```

23) WAP to demonstrate method overloading

⇒
```
class Overload
{
void show()
{
System.out.println("In parameter show");
}
void show(int a)
{
System.out.println("In show with one parameter");
}
}
class Main
{
public static void main(String args[])
{
Overload o=new Overload();

o.void show();
o.void show(70);

}
}
```

Shot on OnePlus
By Yash Bule

24) WAP to demonstrate single inheritance

⇒
```
class A
{
void DisplayA():
{
System.out.println ("In A");
}
}
class B extends A
{
void Display B()
{
System.out.println ("In B");
}
}
class Main
{
public static void main (String args[])
{
B b = new B();

b.DisplayA();
b.DisplayB();

}
}
```

25) WAP to demonstrate Multilevel Inheritance

⇒
```
class A
{
void DisplayA()
{
System.out.println ("In A");
}
}
class B extends A
{
void DisplayB()
{
System.out.println ("In B");
}
}
Class C extends B
{
void DisplayC()
{
System.out.println ("In C");
}
}

class Main
{
public static void main (String args[])
{
C a = new C();
a.DisplayA();
a.DisplayB();
a.DisplayC();
}}
```

Shot on OnePlus
By Yash Bule

26) WAP to demonstrate Hierarchical Inheritance

⇒
```
class A
{
void DisplayA()
{
System.out.println("In A");
}
}

class B extends A
{
void DisplayB()
{
System.out.println("In B");
}
}

class C extends A
{
void DisplayC()
{
System.out.println("In C");
}
}

class Hierarchical
{
public static void main(String args[])
{
C a=new C();

a.DisplayA();
a.DisplayC();
System.out.println(" Calling for B class");

B b=new B();

b.DisplayA();
b.DisplayB();
}
}
```

27) WAP to demonstrate abstract keyword.

⇒
```
abstract class A
{
abstract void show1();
}
void show2()
{
System.out.println("In show 2");
}
class B extends A
{
void show1()
{
System.out.println("In show 1");
}
}
class Main
{
public static void main(String args[])
{
```

```
B b = new B();

b.show1();
b.show2();
}
}
```

27) WAP to demonstrate packages

```
package pack;

import java.io.*;

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

}
```

```
package mypack;

import pack.*;

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

A obj = new A();

obj.msg();
}

}
```

28) WAP to demonstrate interface

=>
```
interface A
{
void showA();
}
~~interface~~ class B implement A
{
void showA()
{
System.out.println("In show");
}
}
class Main
{
public static void main(String args[])
{
B b = new B();

b.showA();

}
}
```

29) WAP to demonstrate Exception Handling (Arithmetic Exception)

=>
```
class Exdemo
{
public static void main(String args[])
{
System.out.println("In Main");

int a, b, c;

a = 100;
b = 0;

try
{
c = a/b;
}
catch (ArithmeticException e)
{
System.out.println(e);
}
System.out.println("Existing Main");
}
}
```

Shot on OnePlus
By Yash Bule

30) WAP to demonstrate Exception Handling. (ArrayIndexOutOfBounds Exception)

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

try
{
int arr[] = new arr[5];

arr[7] = 500;
}
catch(ArrayIndexOutOfBounds Exception e)
{
System.out.println(e);
}
System.out.println("Existing Main");
}
}
```

31) WAP to demonstrate Try with Multiple catch

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

int a, b, c;

a = 100 ; b = 10; c = 0;

try
{
c = a/b ;
int arr[] = new arr[5];
arr[7] = 500 ;
}
catch (Arithmetic Exception e)
{
System.out.println(e);
}
catch (ArrayIndexOutOfBounds Exception e2)
{
System.out.println(e2);
}
System.out.println("c = " + c);

System.out.println("Existing Main");
}
}
```

Shot on OnePlus
By Yash Bule

32) WAP to demonstrate Nested try-catch.

⇒
```
class Excep
{
public static void main (String args[])
{
try
{
try
{
System.out.println (" Going to divide");
int a = 100, b = 0;
int c = a/b
}
catch (Arithmetic Exception e)
{
System.out.println (e);
}
try
{
int a a[] = new a[5];
a[7] = 4;
}
catch (ArrayIndexOutOfBounds Exception e)
{
System.out.println (e);
}
System.out.println (" Other statement");
}
catch (Exception e)
{
System.out.println (" Handled");
}
}
```

33) WAP to demonstrate

⇒
```
class ThdemoI
{
public static void main (String args[])
{
System.out.println (" In Main");
try
{
int a = 100;
if (a == 100)
{
throw new Arithmetic Exception (" value of a is equal");
}
else
{
System.out.println (" a =" + a);
}
catch (Arithmetic Exception e)
{
System.out.println (e);
}
System.out.println (" Existing Main");
}
}
```

Shot on OnePlus
By Yash Bule

34) WAP to demonstrate finally keyword.

⇒
```
class FinalDemo
{
public static void main (String args [])
{
 int a,b,c;
  a=100;
  b=0;
try
{
  c = a/b;
}
catch (ArithmeticException e)
{
System.out.println (e);

System.out.println ("Existing Main");
}
finally
{
System.out.println ("Final block is working");
}
}
```

35) WAP to demonstrate Multithreading using Start.

⇒
```
public class StartEx extends Thread
{
public void run()
{
System.out.println ("Thread is running");
}
public static void main (String args[])
{
 StartEx s1 = new StartEx ();

  s1.start();
}
}
```

36) ~~pul~~ WAP to demonstrate multithread using CurrentThread

⇒
```
class CurrentThread extends Thread
{
public void run()
{
System.out.println(Thread.currentThread().getName());
}
public static void main(String args[])
{

CurrentThread t1 = new CurrentThread();

CurrentThread t2 = new CurrentThread();

t1.start();

t2.start();
}
}
```

37) WAP to demonstrate multithreading using ~~GetPriority~~ GetName

⇒ ~~class Getpriority~~

```
class GetName extends Thread
{
public void run()
{
System.out.println(Thread.currentThread().getName());
}
public static void main(String args[])
{
GetName t1 = new GetName();

GetName t2 = new GetName();

t1.start();

t2.start();
}
}
```

Shot on OnePlus
By Yash Bule

38) WAP to demonstrate multithreading using SetName.

⇒
```
class SetName extends Thread
{
public void run()
{
Thread.currentThread().setName ("Pratik");
System.out.println (Thread.currentThread().getName ());
}
public static void main (String args [])
{
SetName t1 = new SetName();
SetName t2 = new SetName
t1.start();
}
}
```

39) WAP to demonstrate Multithreading using GetPriority.

⇒
```
class GetPriority extends Thread
{
public void run()
{
System.out.println ("Thread is running...");
}
public static void main (String args [])
{
GetPriority t1 = new GetPriority();
System.out.println("GetPriority =" + t1.getPriority ());
t1.start();
}
}
```

40) WAP to demonstrate Multithread using SetPriority

⇒
```
class SetPriority extends Thread
{
public void run()
{
System.out.println("Thread is running ..... ");
}
public static void main(String args[])
{
SetPriority t1 = new SetPriority();
t1.setPriority(1);
t1.set.
System.out.println("SetPriority =" + t1.setPriority());
t1.start();
}
}
```

41) WAP to demonstrate FileOutputStream for 1 byte.

⇒
```
import java.io.FileOutputStream;

public class FileOutput
{
public static void main(String args[])
{ try {
FileOutputStream fout = new FileOutputStream("D:\\Pratik.txt");

fout.write(65);

fout.close();

System.out.println("Success");
}
catch (Exception e)
{
System.out.println(e);
}
}
}
```

42) WAP to demonstrate write all character using FileOutputStream

⇒
```
import java.io.FileOutputStream;

public class FileOutput
{
public static void main (String args[])
{
try
{
FileOutputStream f=new FileOutputStream ("D:\\Pratik.txt");

String s="Welcome to java";

byte b[] = s.getBytes();

f.write(b);

f.close();
System.out.println ("Success");
}
catch (Exception e)
{
System.out.println (e);
}
}
}
```

43) WAP to read single character using FileInputStream

⇒
```
import java.io.FileInputStream;

public class FileInput
{
public static void main (String args[])
{
try
{
FileInputStream f = new FileInputStream ("D:\\Pratik.txt");

int i = f.read();

System.out.println ((char)i);

f.close();
}
catch (Exception e)
{
System.out.println (e);
}
}
}
```

Shot on OnePlus
By Yash Bule

44) WAP to read all character using FileInputStream

⇒

```
import java.io.FileInputStream;

public class FileInput
{
public static void main (String args[])
{
try
{
FileInputStream f=new FileInputStream ("D:\\Pratik.txt");

int i=0

while (i=f.read() != -1)
{
System.out.println ((char) i);
}

f.close();
}
catch (Exception e)
{
System.out.println (e);
}
}
}
```

Shot on OnePlus
By Yash Bule