**1.Write 3 different java programs to print the following patterns**

**a) 1**

**12**

**123**

**1234**

**12345**

Ans--------------------------------------------------------------

public class Test1 {

public static void main(String[] ar)

{

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

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

System.out.print(j);

}

System.out.println();

}

}

}

**b) 54321**

**5432**

**543**

**54**

**5**

Ans-------------------------------------------------------

public class Test1 {

public static void main(String[] ar)

{

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

for(int j=5;j>=i;j--){

System.out.print(j);

}

System.out.println();

}

}

}

**c) x**

**xxx**

**xxxxx**

**xxxxxxx**

**xxxxx**

**xxx**

**x**

**Note: Shape will be Rhombus.**

Ans-----------------------------------------------------------

public class Test {

public static void main(String[] ar)

{int i,sp=3,j,k;

for(i=1;i<=7;i+=2){

for(k=1;k<=sp;k++)

System.out.print(" ");

sp--;

for(j=1;j<=i;j++){

System.out.print("X");

}

System.out.println();

}

sp=1;

for(i=5;i>=1;i-=2){

for(k=1;k<=sp;k++)

System.out.print(" ");

sp++;

for(j=1;j<=i;j++){

System.out.print("X");

}

System.out.println();

}

}

}

**2. Write a java program to take the input from user and determine if it is a prime number or not.**

Ans---------------------------------------

import java.util.Scanner;

public class Test{

public static void main(String[] ar)

{ Scanner in=new Scanner(System.in);

System.out.print("Enter a number : ");

int num=in.nextInt();

boolean prime=true;

for(int i=2;i<=Math.sqrt(num);i++){

if(num%i==0){

prime=false;

break;

}

}

if(prime)

System.out.println("Prime Number");

else

System.out.println("Not Prime Number");

}

}

**3. Write a java program to display the fibonacci series till less than 200 using only 2 variables.**

Ans-------------------------------------

public class Test {

public static void main(String[] ar)

{

int a=0,b=1;

System.out.println("0\n1");

while((a+b)<200){

System.out.println(a+b);

b=a+b;

a=b-a;

}

}

}

**5.Write Java program to check if a name is palindrome.**

Ans--------------------------------------------------------------

import java.util.Scanner;

public class Test {

public static void main(String[] ar)

{ Scanner in=new Scanner(System.in);

System.out.print("Enter word : ");

String name=in.next();

boolean pal=true;

int len=name.length();

for(int i=0;i<len/2;i++){

if(name.charAt(i)!=name.charAt(len-1-i)){

pal=false;

break;

}

}

if(pal)

System.out.println("Palindrom");

else

System.out.println("Not Palindrom");

}

}

**6.Write Java program to check if a number is Armstrong number or not? (input 153 output true, 123 output false)**

Ans---------------------------------------

import java.util.Scanner;

public class Test {

public static void main(String[] ar)

{ Scanner in=new Scanner(System.in);

System.out.print("Enter a number : ");

int num=in.nextInt();

int temp1=num,temp2=0;

while(temp1!=0){

temp2=temp2+(temp1%10)\*(temp1%10)\*(temp1%10);

temp1=temp1/10;

}

if(temp2==num)

System.out.println(true);

else

System.out.println(false);

}

}

**7.How to find factorial of number in Java using iteration?**

Ans---------------------------------------------

import java.util.Scanner;

public class Test1 {

public static void main(String[] ar)

{ Scanner in=new Scanner(System.in);

System.out.print("Enter a number : ");

int num=in.nextInt();

int fact=1;

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

fact=fact\*i;

}

System.out.println(fact);

}

}

**8.Write a Java code to take a character as a input from user and determine if it is a vowel or a consonant using conditional construct.**

import java.util.Scanner;

public class Test1 {

public static void main(String[] ar)

{ Scanner in=new Scanner(System.in);

System.out.print("Enter a character : ");

char alphabet=in.next().charAt(0);

switch(alphabet){

case 'a':

case 'A':

System.out.println("Vowel");

break;

case 'e':

case 'E':

System.out.println("Vowel");

break;

case 'i':

case 'I':

System.out.println("Vowel");

break;

case 'o':

case 'O':

System.out.println("Vowel");

break;

case 'u':

case 'U':

System.out.println("Vowel");

break;

default:

System.out.println("Consonant");

}

}

}

**9. Write a switch case java code to create calculator with + - / \* functionalities only.**

import java.util.Scanner;

public class Test1 {

public static void main(String[] ar)

{ Scanner in=new Scanner(System.in);

System.out.print("Enter First Number : ");

int a=in.nextInt();

System.out.print("Enter Second Number : ");

int b=in.nextInt();

System.out.print("Enter operation (+,-,\*,/) : ");

char op=in.next().charAt(0);

switch(op){

case '+':

System.out.println("Sum is "+(a+b));

break;

case '-':

System.out.println("Subtraction is "+(a-b));

break;

case '\*':

System.out.println("Multiplication is "+(a\*b));

break;

case '/':

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

break;

default:

System.out.println("Invalid operation");

}

}

}

**10. Write a java code to copy one array into another.**

public class Test{

public static void main(String[] ar)

{ Scanner in=new Scanner(System.in);

int first[]={12,4,56,23,67,89,48,78};

int second[]=new int[first.length];

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

second[i]=first[i];

System.out.print("first : ");

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

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

System.out.print("\nsecond : ");

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

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

}

}

**11. Write a java code to compare the length of two arrays and display the longer array.**

public class Test{

public static void main(String[] ar){

int first[]={1,2,3,4,5,6,7};

int second[]={1,2,3,4,5,6,7,8};

if(first.length>second.length){

System.out.print("First array is greater : ");

for(int x:first){

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

}

}

else{

System.out.print("Second array is greater : ");

for(int x:second){

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

}

}

}

}

**12. Write a java code to display a reverse String array.**

public class Test{

static void stringReverse(String input[]){

int j;

String temp;

for(j=0;j<input.length/2;j++){

temp=input[j];

input[j]=input[input.length-j-1];

input[input.length-j-1]=temp;

}

}

public static void main(String[] ar){

String strings[]={"India","Australia","America","Africa","London","Brazil"};

System.out.print("Input : ");

for(String s:strings){

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

}

System.out.println();

stringReverse(strings);

System.out.print("Output : ");

for(String s:strings){

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

}

}

}

**13. Write the difference between checked and unchecked exception with example code**

**Ans. Checked Exception** are the exceptions that are checked at compile time. If some code throws a checked exception, then it must handle the exception by surrounding try catch.

Those exceptions that are not checked at compiled time are called Unchecked Exception. All RuntimeException classes are unchecked exceptions.

Ex: IOException is checked exception and ArithmeticException is unchecked exception

import java.io.FileInputStream;

import java.io.IOException;

public class Test1{

public static void main(String ar[]){

try{ int i=0;

FileInputStream fin=new FileInputStream("myfile.txt");

while((i=fin.read())!=-1){

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

}

fin.close();

}catch(IOException e){

System.out.println(45/0);//Arithmetic exception

}

}

}

**14. Write the difference between throw and throws with example code**

Ans. Throw keyword is used to explicitly throw an exception while throws keyword is used to declare that a method can throw a exception.

**15. Write a note or nested try…catch block with example code**

**Ans.**

**16. Write a note on MultiThreading and MultiTasking**

**Ans.** MultiThreading is running different operation concurrently in a single process. while Multitasking is Multitasking is sharing of computer resources like CPU,RAM etc among processes.

**17. Write a short note on Deque and give example code.**

**Ans.** Deque Interface is a linear collection that supports element insertion and removal at both ends. It internally use double ended queue data structure to the elements.

import java.util.\*;

public class Test1 {

public static void main(String[] args) {

Deque<String> que = new ArrayDeque<String>();

que.add("Vijay");

que.add("Ajay");

que.add("Sanjay");

for (String s : que) {

System.out.println(s);

}

}

}

**18. Write a short note on Generics an all types of Parameters used in Generics with example code.**

Ans. Generics programming is used to achieve type safety in objects. Generics can be used at two places:

a) Method : If you declare any method generic it can be called with arguments of different types.

public class Test {

<E> void display(E x){

System.out.println(x);

}

public static void main(String[] ar)

{ Test ob=new Test();

ob.display("Hello World");

ob.display(45.67f);

ob.display(34.67);

ob.display('H');

ob.display(45);

ob.display(true);

}

}

b) Class : If you declare a class generic its members can be of any type.

class MyGeneric<E>{

private E value;

void setValue(E value){

this.value=value;

}

E getValue(){

return value;

}

}

public class Test1{

public static void main(String[] ar)

{

MyGeneric<Integer> ob=new MyGeneric<>();

ob.setValue(34);

System.out.println(ob.getValue());

MyGeneric<String> ob1=new MyGeneric<>();

ob1.setValue("Hello");

System.out.println(ob1.getValue());

}

}

**19. Write a short note on Map Interface.**

**Ans.** Map is special type of collection which store elements in pair of key and value. Map is not the part of collection hierarchy it has its own hierarchy.

There are three main sub class of map interface:

1. HashMap : It doesn't preserve insertion order.

2. LinkedHashMap : It preserve insertion order.

3. TreeMap : It store elements in sorted order base on keys.

**20. Write the difference between LinkedList and ArrayList.**

Ans. ArrayList internally uses **dynamic array** to store the elements. While LinkedList internally uses **doubly linked list** to store the elements.

Insertion and Deletion in ArrayList is slow but fast in LinkedList.

Accessing data in ArrayList is faster than LinkedList.

**21. Write a note on Dynamic array in java.**

Ans. Dynamic array a special type of array which can grow and shrink their size as per the requirement. In java ArrayList internally uses the dynamic array it can grow its size when you add element in the list.

**22. What is the purpose of the System class?**

Ans. System class is used for standard input/output and error output stream.

Standard Fields of System class are:

in : for standard input

our : for standard output

err :for error output stream

**23. Which is the abstract parent class of FileWriter ?**

**Ans.** OutputStreamWriter

**24. Which class is used to read streams of characters from a file?**

**Ans.** FileReader

**25. Which class is used to read streams of raw bytes from a file?**

**Ans.** FileInputStream

**26. What are the differences between FileInputStream/FileOutputStream and RandomAccessFile**

**Ans.** FileInputStream is used to read stream of bytes from a file.

FileOutputStream is used to write stream of bytes from a file.

**27. Write a note on Channels and Buffer with example.**

**28. What is the difference between System.out ,System.err and System.in?**

**Ans. System.out**  is used for standered output.

Ex. System.out.println() :- it is used to print something on console.

**System.err** is used for error output stream

Ex. System.err.println() :- It is used to show error messages while debugging program.

**System.in** is used for standard input.

It is used to take input from console.

**29. What is the purpose of the System class?**

Ans. System class is used for standard input/output and error output stream.

Standard Fields of System class are:

in : for standard input

our : for standard output

err :for error output stream

**30. Which is the abstract parent class of FileWriter ?**

**Ans.** OutputStreamWriter

**31. Which class is used to read streams of characters from a file?**

import java.io.\*;

public class Test1{

public static void main(String args[])throws Exception{

FileReader fread=new FileReader("myfile.txt");

BufferedReader bread=new BufferedReader(fread);

String l;

while ((l = bread.readLine()) != null) {

System.out.println(l);

}

bread.close();

fread.close();

}

}

**32. Which class is used to read streams of raw bytes from a file?**

**Ans--------------------------------**

import java.io.FileInputStream;

public class Test{

public static void main(String ar[]){

try{ int i=0;

FileInputStream fin=new FileInputStream("myfile.txt");

while((i=fin.read())!=-1){

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

}

fin.close();

}catch(Exception e){

System.out.println(e);

}

}

}

**33. What are the differences between FileInputStream/FileOutputStream and RandomAccessFile**

**Ans.** FileInputStream is used to read stream of bytes from a file.

FileOutputStream is used to write stream of bytes from a file.

Random access file is a special kind of file in Java which allows non-sequential or random access to any location in file.

34. Write a note on Channels and Buffer with example.

**35. Write a note on PreparedStatement and ResultSetMetaData interfaces with code snippets.**

**Ans.** PreparedStatement is used to pass parameter to query at runtime and ResultSetMetaData is used to get structure details of tables.

import java.sql.\*;

class Test1{

public static void main(String args[]){

Class.forName("org.h2.Driver");

String url="jdbc:h2:~/mydb";

String username="root";

String password="root";

try(Connection con=DriverManager.getConnection(url,username,password);

PreparedStatement ps=con.prepareStatement("select \* from stu");

ResultSet rs=ps.executeQuery()){

ResultSetMetaData rsmd=rs.getMetaData();

System.out.println("No of columns: "+rsmd.getColumnCount());

for(int i=1;i<=rsmd.getColumnCount();i++){

System.out.println("Column Name of 1st column: "+rsmd.getColumnName(i));

System.out.println("Column Type Name of 1st column: "+rsmd.getColumnTypeName(i));

}

}catch(Exception e){

System.out.println(e);

}

}

}

**36. Write a note on DDL, DML, DQL, DDL with code snippets.**

**DDL (Data Definition Language)**

DDL is used to add / modify / delete the structures of table and database.

It consists CREATE, ALTER and DROP statements.

* create table stu(id int,name varchar(20),subject varchar(20));
* ALTER TABLE stu RENAME COLUMN id to roll;
* DROP TABLE stu;

**DML (Data Manipulation Language)**

DML is used to add / modify / delete data itself.

It consists INSERT, UPDATE and DELETE statements.

* INSERT INTO stu VALUES(23,'Name','English');
* UPDATE stu SET roll=45 WHERE name='Name';
* DELETE FROM stu WHERE name='Name';

**DQL (Data Query Language)**

SELECT is the main DQL instruction. It retrieves data you need. SHOW retrieves information about the metadata.

It consists SELECT, SHOW statements.

* SELECT \* FROM stu;
* SHOW TABLES;

**37. Write a note on HTML , CSS and Javascript.**

**Ans. HTML :** HTML stands for HyperText Markup Language. It is used to create Web pages.

**CSS :**  CSS stands for Cascading Style Sheets. It is used add styling to the web pages.

**JavaScript :** JavaScript is object oriented programming language used to add programming features to web pages.

**38. Write a code to fetch the data from H2 and put it in any collection object and display it.**

import java.sql.\*;

import java.util.\*;

class Student{

int roll;

String name,subject;

public String toString(){

return roll+" : "+name+" : "+subject;

}

}

public class Test{

public static void main(String args[]){

Student s;

List<Student> list=new ArrayList<>();

Class.forName("org.h2.Driver");

String url="jdbc:h2:~/mydb";

String username="root";

String password="root";

try(Connection con=DriverManager.getConnection(url,username,password);

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select \* from student")){

while(rs.next()){

s=new Student();

s.roll=rs.getInt("roll");

s.name=rs.getString("name");

s.subject=rs.getString("subject");

list.add(s);

}

for(Student x:list){

System.out.println(x);

}

}catch(Exception e){

System.out.println(e);

}

}

}

**39. Describe the different approaches of String processing.**

**Ans.** String class : It is used to create a immutable string.

StringBuffer class : It is used to create mutable string and it is synchronized

StringBuilder class : It is used to create mutable string and it is not synchronized

**40. What is the difference between System.out ,System.err and System.in?**

**Ans. System.out**  is used for standered output.

Ex. System.out.println() :- it is used to print something on console.

**System.err** is used for error output stream

Ex. System.err.println() :- It is used to show error messages while debugging program.

**System.in** is used for standard input.

It is used to take input from console.

**41. What is the purpose of the System class?**

Ans. System class is used for standard input/output and error output stream.

Standard Fields of System class are:

in : for standard input

our : for standard output

err :for error output stream

**42. Which is the abstract parent class of FileWriter ?**

**Ans.** OutputStreamWriter

**43. Which class is used to read streams of characters from a file?**

import java.io.\*;

public class Test1{

public static void main(String args[])throws Exception{

FileReader fread=new FileReader("myfile.txt");

BufferedReader bread=new BufferedReader(fread);

String l;

while ((l = bread.readLine()) != null) {

System.out.println(l);

}

bread.close();

fread.close();

}

}

**44. Which class is used to read streams of raw bytes from a file?**

**Ans--------------------------------**

import java.io.FileInputStream;

public class Test{

public static void main(String ar[]){

try{ int i=0;

FileInputStream fin=new FileInputStream("myfile.txt");

while((i=fin.read())!=-1){

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

}

fin.close();

}catch(Exception e){

System.out.println(e);

}

}

}

**45. What are the differences between FileInputStream/FileOutputStream and RandomAccessFile**

**Ans.** FileInputStream is used to read stream of bytes from a file.

FileOutputStream is used to write stream of bytes from a file.

Random access file is a special kind of file in Java which allows non-sequential or random access to any location in file.

**46) Write a note on Channels and Buffer with example.**

Ans) Channel implementation uses the native code to perform actual work. The channel interface allows us to gain access to low-level I/O services in a portable and controlled way.

In Java NIO the primary Channels used are given below:

* **FileChannel:** The file channel is used for reading the data from the files. It's object can be created only by calling the getChannel() method. We cannot create FileChannel object directly.

Let's see the example to create the object of FileChannel:

* 1. FileInputStream fis = **new** FileInputStream("D:\\testin.txt"); // Path of Input text file
  2. ReadableByteChannel rbc = fis.getChannel();
* **DatagramChannel:** The datagram channel can read and write the data over the network via UDP (User Datagram Protocol). It uses the factory methods for creating the new object.

The syntax used for opening the DatagramChannel:

* 1. DatagramChannel ch = DatagramChannel.open();

The syntax used for closing the DatagramChannel:

* 1. DatagramChannel ch = DatagramChannel.close();
* **SocketChannel:** The datagram channel can read and write the data over the network via TCP (Transmission Control Protocol). It also uses the factory methods for creating the new object.

The syntax used for opening the SocketChannel:

* 1. SocketChannel ch = SocketChannel.open();
  2. ch.connect(**new** InetSocketAddress("somehost", someport));

The syntax used for closing the SocketChannel:

* 1. SocketChannel ch = SocketChannel.close();
  2. ch.connect(**new** InetSocketAddress("somehost", someport));
* **ServerSocketChannel:** The ServerSocketChannel allows user to listen the incoming TCP connections, same as a web server. For every incoming connection a SocketChannel is created.

The syntax used for opening the ServerSocketChannel:

* 1. ServerSocketChannel ch = ServerSocketChannel.open();
  2. ch.socket().bind (**new** InetSocketAddress (somelocalport));

The syntax used for closing the ServerSocketChannel:

* 1. ServerSocketChannel ch = ServerSocketChannel.close();
  2. ch.socket().bind (**new** InetSocketAddress (somelocalport));