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

**a) 1**

**12**

**123**

**12345**

a)importjava.util.\*;

classPattern\_DT

{

public static void main(String ab[])

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter value");

int v=sc.nextInt();

for(inti=1;i<=v;i++)

{

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

System.out.print(j);

System.out.println();

}

}

}

**b) 54321**

**5432**

**543**

**54**

**5**

b)importjava.util.\*;

classPattern\_DT

{

public static void main(String ab[])

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter value");

int v=sc.nextInt();

for(inti=1;i<=v;i++)

{

for(int j=v;j>=i;j--)

System.out.print(j);

System.out.println();

}

}

}

**c)     x**

**xxx**

**xxxxx**

**xxxxxxx**

**xxxxx**

**xxx**

**x**

**Note: Shape will be Rhombus.**

c)importjava.util.\*;

classPattern\_DT

{

public static void main(String ab[])

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter N : ");

int n=sc.nextInt();

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

char c = sc.next().charAt(0);

for(inti=1;i<=n;i++)

{

for(int j=1;j<=n-i;j++)

{

System.out.print(" ");

}

for(int j=1;j<=i\*2-1;j++)

{

System.out.print(c);

}

System.out.println();

}

for(inti=n-1;i>0;i--)

{

for(int j=1;j<=n-i;j++)

{

System.out.print(" ");

}

for(int j=1;j<=i\*2-1;j++)

{

System.out.print(c);

}

System.out.println();

}

}

}

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

importjava.util.\*;

classPattern\_DT

{

public static void main(String ab[])

{

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int count=0;

for(inti=1;i<=n;i++)

{

if(n%i==0)

{

count++;

}

}

if(count==2)

System.out.println(n+" is a prime");

else

System.out.println(n + " is not a prime");

}

}

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

importjava.util.\*;

classPattern\_DT

{

public static void main(String ab[])

{

int a=0;

int b=1;

System.out.println(a);

System.out.println(b);

while(a<200 && b<200)

{

a=a+b;

if(a<200)

System.out.println(a);

b=a+b;

if(b<200)

System.out.println(b);

}

}

}

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

importjava.util.\*;

classPattern\_DT

{

public static void main(String ab[])

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter value");

String v=sc.next();

inti,j;

charch[]=v.toCharArray();

for(i=ch.length-1,j=0;i>=j;)

{

if(ch[i]==ch[j])

{

j++;

i--;

}

else

break;

}

if(i<j)

System.out.println("String is Palindrome");

else

System.out.println("Not a Palindrome");

}

}

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

 importjava.util.\*;

classPattern\_DT

{

public static void main(String ab[])

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter value");

int v=sc.nextInt();

int t=v;

int v1,num=0;

while(v>0)

{

v1=v%10;

num=num+(v1\*v1\*v1);

v=v/10;

}

if(num==t)

System.out.println("true ");

else

System.out.println("false");

}

}

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

 importjava.util.\*;

classPattern\_DT

{

staticint fact(int n)

{

int p=1;

for(inti=1;i<=n;i++)

p=p\*i;

return p;

}

public static void main(String ab[])

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter value");

int v=sc.nextInt();

System.out.println(" Factorial number : "+fact(v));

}

}

**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.**

importjava.util.\*;

classPattern\_DT

{

public static void main(String ab[])

{

Scanner sc=new Scanner(System.in);

char c;

System.out.println("Enter value");

c=sc.next().charAt(0);

if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u')

System.out.println("Vowel");

else

System.out.println("Consonant");

}

}

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

importjava.util.\*;

classPattern\_DT

{

public static void main(String ab[])

{

Scanner sc=new Scanner(System.in);

char c;

do

{

System.out.println("Enter value");

int a=sc.nextInt();

c=sc.next().charAt(0);

int b=sc.nextInt();

switch(c)

{

case '+':

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

break;

case '-':

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

break;

case '\*':

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

break;

case '/':

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

break;

}

}while(c!='@');

}

}

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

importjava.util.\*;

classPattern\_DT

{

public static void main(String ab[])

{

int a[]={1,2,3,67,89};

int b[]=new int[a.length];

b=a;

for(int c:b)

System.out.println(c);

}

}

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

importjava.util.\*;

classPattern\_DT

{

public static void main(String ab[])

{

int a[]={1,2,3,67,89};

int b[]={11,22,33};

if(a.length>b.length)

{

System.out.println("Longer Array is A");

for(intaa:a)

System.out.println(aa);

}

else

{

System.out.println("Longer Array is B");

for(intbb:b)

System.out.println(bb);

}

}

}

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

importjava.util.\*;

classPattern\_DT

{

public static void main(String ab[])

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter size");

int size=sc.nextInt();

String str[]=new String[size];

System.out.println("Enter value");

for(inti=0;i<str.length;i++)

str[i]=sc.next();

System.out.println("Reverse value");

for(inti=str.length-1;i>=0;i--)

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

}

}

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

Checked: are the exceptions that are checked at compile time. If some code within a method throws a checked exception, then the method must either handle the exception or it must specify the exception using throws keyword.

Unchecked are the exceptions that are not checked at compiled time. In C++, all exceptions are unchecked, so it is not forced by the compiler to either handle or specify the exception.

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

|  |  |  |
| --- | --- | --- |
| No. | throw | throws |
| 1) | Java throw keyword is used to explicitly throw an exception. | Java throws keyword is  used to declare an exception. |
| 2) | Checked exception cannot be propagated using throw only. | Checked exception can  be propagated with throws. |
| 3) | Throw is followed by an instance. | Throws is followed by class. |
| 4) | Throw is used within the method. | Throws is used with the  method signature. |
| 5) | You cannot throw multiple exceptions. | You can declare multiple  exceptions . |

void m()throws ArithmeticException{

throw new ArithmeticException("sorry");

}

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

The try block within a try block is known as nested try block in java.

class Excep6{

 public static void main(String args[]){

  try{

    try{

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

     int b =39/0;

    }catch(ArithmeticException e){System.out.println(e);}

    try{

    int a[]=new int[5];

    a[5]=4;

    }catch(ArrayIndexOutOfBoundsException e){System.out.println(e);}

    System.out.println("other statement);

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

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

}

}

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

**MultiTasking**

Multitasking means an OS can have multiple processes(tasks) run simultaneously, in whatever manner it wants.

Multitasking is sharing of computing resources(CPU, memory, devices, etc.) among processes

**MultiThreading :**

Multithreading means in any single process, multiple threads is allowed and again, can run simultaneously.

Multithreading is sharing of computing resources among threads of a single process.

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

Java Deque Interface is a linear collection that supports element insertion and removal at both ends. Deque is an acronym for "double ended queue".

class Sample

{

public static void main(String ab[])  
 {

java.util.Deque<String> d=new java.util.ArrayDeque<>();

d.add("aaa");

d.add("bbb");

d.add("ccc");

d.offerFirst("eee");

for(String a:d)

System.out.println(a);  
 }  
}

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

Generics mainly used to improve type safety.

class Sample<T>

{

T a;

Sample(T a)

{

this.a=a;  
 }

void dis()

{

System.out.println(a);  
 }

public static void main(String ab[])

{

Sample<Integer> s1=new Sample<>(10);

s1.dis();

Sample<Character> s2=new Sample<>('a');

s2.dis();

Sample<String> s3=new Sample<>("hai");

s3.dis();

}  
}

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

Map stores the data as Key-Value<K,V> Pairs.

It does not allow Duplicate Key Values

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

|  |  |
| --- | --- |
| LinkedList | ArrayList |
| LinkedList internally uses doubly linked list to store the elements. | ArrayList internally uses dynamic array to store the elements. |
| Manipulation with LinkedList is faster than ArrayList because it uses doubly linked list so no bit shifting is required in memory. | Manipulation with ArrayList is slow because it internally uses array. If any element is removed from the array, all the bits are shifted in memory. |
| LinkedList is better for manipulating data. | ArrayList is better for storing and accessing data. |

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

A *dynamic array* has variable size and allows elements to be added or removed.

A dynamic array, growable array, resizable array, dynamic table, mutable array, or array list is a random access, variable-size list data structure that allows any number of elements to be added or removed.

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

The purpose of the System class is to provide access to system resources.

System class is provided with useful fields (static members) pertaining to the environment.  
  
Standard input,output and error output streams are provided with System class. These are used to access the externally defined properties and environment variables.

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

OutputStreamWriter is a child for Writer

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

FileReader

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

InputStream(FileInputStream)

**26.   What are the differences between FileInputStream/FileOutput**

**Stream and RandomAccessFile**

RandomAccessFile treats the file as an array of bytes where it has the internal pointer. The fact that it treats it like a large array of bytes

FileInputStream reads the stream and returns the data. It is more suited to reading raw data like images etc. It does not treat the file as a large array, it just keeps tabs of where in the file it has read so far.

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

Buffers provide a mechanism to store a fixed amount of primitive data elements in an in-memory container. In the NIO, all data is handled with buffers. When data is read, it is read directly into a buffer. When data is written, it is written into a buffer.

Buffers work with channels. Channels are portals through which I/O transfers take place, and buffers are the sources or targets of those data transfers.

|  |  |  |
| --- | --- | --- |
| import java.io.file.Paths; | | |
|  | import java.nio.file.Path; | |
|  | import java.io.IOException; | |
|  | import java.nio.ByteBuffer; | |
|  | import java.nio.channels.FileChannel; | |
|  | import java.io.FileReader; | |
|  | import java.io.BufferedReader; | |
|  |  | |
|  | public class FileChannelRead { | |
|  | public static void main (String [] args) | |
|  | throws Exception { | |
|  |  | |
|  | new FileChannelRead().readFile(); | |
|  | } | |
|  | private void readFile() | |
|  | throws IOException { | |
|  |  | |
|  | String filePath = "readfile.txt"; | |
|  | printFileContents(filePath);  Path path=Paths.get(filePath); | |
|  | |
|  | |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  | FileChannel fileChannel =  FileChannel.open(path); |
|  | ByteBuffer buffer = ByteBuffer.allocate(6); |
|  | int noOfBytesRead = fileChannel.read(buffer); |
|  |  |

|  |  |  |
| --- | --- | --- |
|  | while (noOfBytesRead != -1) { | |
|  | System.out.println("Number of bytes read: " + noOfBytesRead); | |
|  | buffer.flip(); | |
|  | System.out.print("Buffer contents: "); | |
|  |  | |
|  | while (buffer.hasRemaining()) { | |
|  | System.out.print((char) buffer.get()); | |
|  | } | |
|  |  | |
|  | System.out.println(" "); | |
|  | buffer.clear(); | |
|  | noOfBytesRead = fileChannel.read(buffer); | |
|  | } | |
|  | fileChannel.close(); | |
|  | | } | |

}

The output is:

|  |  |
| --- | --- |
|  | File contents: 1234567890 |
|  | Number of bytes read: 6 |

|  |  |
| --- | --- |
|  | Buffer contents: 123456 |
|  | Number of bytes read: 4 |

|  |  |
| --- | --- |
|  | Buffer contents: 7890 |

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

in stream carries (flows) data from Keyboard to CPU. out stream carries data from CPU to console (command prompt). err stream is used by OS to display error messages while the OS is getting booted.

 Java connects to these streams through System class as System.in, System.out and System.err.

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

The PreparedStatement interface is a subinterface of Statement. It is used to execute parameterized query.It is used for dynamic query

PreparedStatement ps=con.prepareStatement("insert into student values(?,?)");

ps.setInt(1,120);

ps.setString(2,"qqq");

ps.executeUpdate();

The metadata means data about data i.e. we can get further information from the data.

If you have to get metadata of a table like total number of column, column name, column type etc. , ResultSetMetaData interface is useful because it provides methods to get metadata from the ResultSet object.

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

ResultSet rs=ps.executeQuery();

ResultSetMetaData rsmd=rs.getMetaData();

System.out.println("Total columns: "+rsmd.getColumnCount());

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

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

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

**Data Definition Language (DDL)**

**Create**

Create command is used to create database and its Objects like tables, index, stored procedure, views , triggers, functions and etc.

eg:create table student(id int,name varchar(20));

**Alter**  
Alter command is used to create database and its Objects

.

eg:alter table student add dob date

**Drop**  
Drop command is used to delete objects from database.

eg:drop table student

**Truncate**  
Trunctae Table command is used to remove all records from a table, including all spaces allocated for records are removed.

eg:truncate table student

**Data Manipulation Language (DML)**  
The commands of SQL that are used to insert data into the database, modify the data of the database and to delete data from the database are collectively called as DML. Examples include Insert, Update and Delete.

**Insert**  
To insert date into a table.

eg:insert into student values(111,'aaa');  
  
**Update**  
To update the existing data in a table.

eg: update student set name='bbb' where id=111  
**Delete**  
delete all records from a table.

delete from student;

**Data Query Language (DQL)**  
The commands of SQL that are used to retrieve data from the database are collectively called as DQL. So all Select statements comes under DQL.  
  
**Select**  
To retreive data from the database table.

eg:select \* from student;

**Data Control Language (DCL)**  
The commands of SQL that are used to control the access to data stored in the database are collectively called as DCL and examples include Grant and Revoke.  
  
**Grant**  
All users access previleges to database.  
  
**Revoke**  
Withdraw users access previleges given by using the Grant command.

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

HTML (Hypertext Markup Language) is the set of [markup](http://searchsoa.techtarget.com/definition/markup) symbols or codes inserted in a file intended for display on a World Wide Web [browser](http://searchwindevelopment.techtarget.com/definition/browser) page. The markup tells the Web browser how to display a Web page's words and images for the user.

A cascading style sheet (CSS) is a Web page derived from multiple sources with a defined order of precedence where the definitions of any style element conflict

JavaScript is a Client side Scripting Language.

Javascript (JS) is a scripting languages, primarily used on the Web. It is used to enhance HTML pages and is commonly found embedded in HTML code. JavaScript is an interpreted language.

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

Student.java

package h2connect;

public class student {

int id;

String name;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

Studentmain.java

package h2connect;

import java.sql.\*;

import java.util.\*;

public class studentmain {

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

Connection con=DriverManager.getConnection("jdbc:h2:tcp://localhost/~/sample","sa","");

Statement st1=con.createStatement();

ResultSet rs=st1.executeQuery("select \* from student");

List<student> al=new ArrayList<>();

while(rs.next())

{

student st=new student();

st.setId(rs.getInt(1));

st.setName(rs.getString(2));

al.add(st);

}

for(student ss:al)

System.out.println(ss.getId()+"\t"+ss.getName());

}

}

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

Generally, string is a sequence of characters. But in java, string is an object that represents a sequence of characters. The java.lang.String class is used to create string object.

How to create String object?

|  |
| --- |
| There are two ways to create String object:  By string literal  By new keyword |