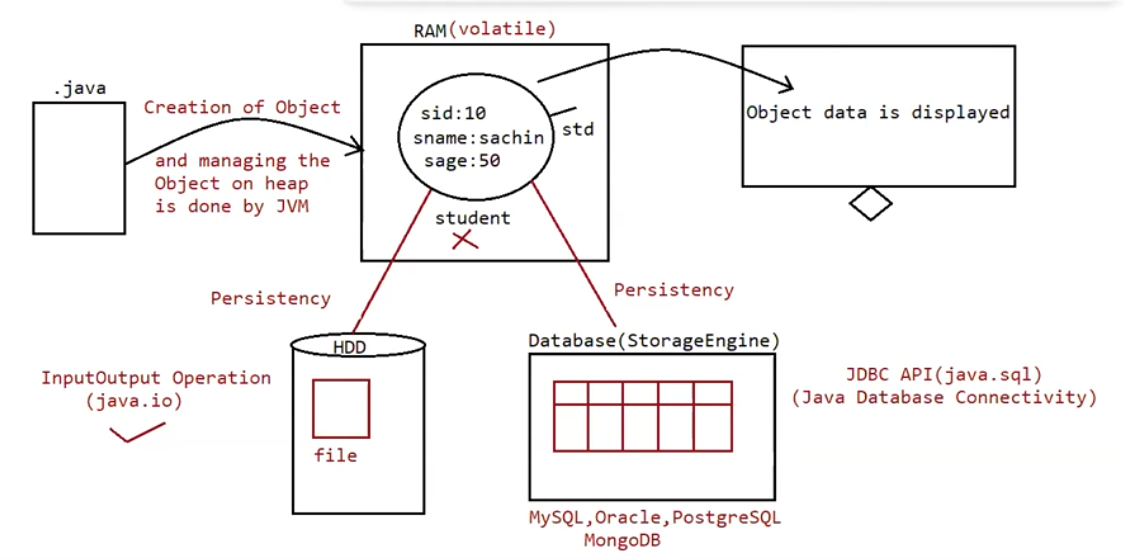
Eg: Io\_Introduction



What is persistency?

It is mechanism of storing the data permanently into a file.

In java persistency can be achieved through an API‘s available inside the package called “java.io”.

Eg: Io\_File\_Creation

File:

File f = new File("abc.txt");

This line 1st checks whether abc.txt file is already available (or) not if it is already available then "f" simply refers that file.

If it is not already available then it won't create any physical file just creates a java File object represents name of the file.

Eg: Io\_Directory\_Creation

Note:

In unix everything is file , java “file IO” is based on unix operating system

Hence in java we can represent both file and directory by file object only.

File class constructors :

1. File f=new File(String name);

=> Creates a java File object that represents name of the file or directory in current working directory.

eg#1. File f=new File("abc.txt");

2. File f=new File(String subdirname,String name);

=> Creates a File object that represents name of the file or directory present in specified sub directory.

eg#1. File f1=new File("abc");

f1.mkdir();

File f2=new File("abc","demo.txt");

Eg: Io\_File\_Inside\_Directory

Eg: Io\_File\_Inside\_Directory\_Eg1

Eg: Io\_Operations\_On\_File

Eg: Io\_Operations\_On\_File\_Eg1

Eg: Io\_FileWriter

FileWriter:

By using FileWriter object we can write character data to the file.

Constructors:

FileWriter fw = new FileWriter(String name);

FileWriter fw = new FileWriter(File f);

The above 2 constructors meant for overriding the data to the file.

Instead of overriding if we want append operation then we should go for the following 2 constructors.

FileWriter fw = new FileWriter(String name,boolean append);

FileWriter fw= new FileWriter(File f,boolean append);

If the specified physical file is not already available then these constructors will create that file.

Methods:

1. write(int ch);

To write a single character to the file.

2. write(char[] ch);

To write an array of characters to the file.

3. write(String s);

To write a String to the file.

4. flush();

To give the guarantee the total data include last character also written to the file.

5. close();

To close the stream.

Note:

=> The main problem with FileWriter is we have to insert line separator manually, which is difficult to the programmer. ('\n')

=> And even line separator varing from system to system.

Eg: Io\_FileWriter\_Eg2

FileReader:

=> By using FileReader object we can read character data from the file.

Constructors:

FileReader fr=new FileReader(String name);

FileReader fr=new FileReader (File f);

Methods

1. int read();

It attempts to read next character from the file and return its Unicode value. If

the next character is not available then we will get -1.

2. int i = fr.read();

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

As this method returns unicode value, while printing we have to perform type

casting.

4. int read(char[] ch);

It attempts to read enough characters from the file into char[] array and returns

the no of characters copied from the file into char[] array.

5. File f=new File("abc.txt");

6. Char[] ch=new Char[(int)f.length()];

7. void close();

Eg: File\_Reader