when you dont override hashcode and equals (given by Object)

Device [deviceId=103, deviceName=Memory Card], Hashcode -> 2018699554

Device [deviceId=107, deviceName=Pen], Hashcode -> 1550089733

Device [deviceId=107, deviceName=Pen], Hashcode -> 865113938

Device [deviceId=107, deviceName=Pen], Hashcode -> 118352462

Device [deviceId=101, deviceName=Marker], Hashcode -> 366712642

Device [deviceId=102, deviceName=Monitor], Hashcode -> 1829164700

Device [deviceId=104, deviceName=Mobile], Hashcode -> 1311053135

Device [deviceId=107, deviceName=Pen], Hashcode -> 1442407170

after overrding hashcode

Device [deviceId=107, deviceName=Pen], Hashcode -> 80

Device [deviceId=107, deviceName=Pen], Hashcode -> 80

Device [deviceId=107, deviceName=Pen], Hashcode -> 80

Device [deviceId=107, deviceName=Pen], Hashcode -> 80

Device [deviceId=101, deviceName=Marker], Hashcode -> 77

Device [deviceId=102, deviceName=Monitor], Hashcode -> 77

Device [deviceId=103, deviceName=Memory Card], Hashcode -> 77

Device [deviceId=104, deviceName=Mobile], Hashcode -> 77

after overrding hashcode & equals

Device [deviceId=107, deviceName=Pen], Hashcode -> 80

Device [deviceId=101, deviceName=Marker], Hashcode -> 77

Device [deviceId=102, deviceName=Monitor], Hashcode -> 77

Device [deviceId=103, deviceName=Memory Card], Hashcode -> 77

Device [deviceId=104, deviceName=Mobile], Hashcode -> 77

List

ArrayList

Vector

Stack (LIFO) - try

LinkedList - try

Set (Unique)

HashSet -> the storage is always in random fashion

they help to store the data fast

But retrival is slower

Stroring user defined object by overriding

hascode and equals

TreeSet -> The storage is always in sorted fashion

they help to retrive data fast

but slower to store

[Laptop, Mobile, Monitor, Pencil]

try to get in descending order

Map<K,V> - in set if you have the duplicate values then new values are not

stored, but in map if the duplicate values are kept then it will

overrite existing value

http://amazon.com

marker 3

mobile 1

HashMap

TreeMap

Hello

Hello

Namaste

Bonjour

Hallo

bye

English - Bye

Hindi - Namste

German - Tschüss

Fresh - au revoir

(5-7)

class Word{

private String language;

private String meaning;

}

Map<String, List<String>>

Map<String, List<Word>>

Enter the word : bye <enter>

Enter the word : Hello <enter>

---------------------------------------------------------------------------

Exception

---------------------------------------------------------------------------

10 / 0

int arr[] = new int[4];

0-3

arr[2] = 33;

arr[3] = 44;

arr[4] = 55; (does not work) -> exception

int arr[] = new int[-4];

java file (exception)

java DB

Exception

Unchecked Exception

are those exceptions which can be handled by JVM/JAVA

Checked Exception

are those exceptions which has to be handled by developer/user

try{

// Business Logics

}catch(ArithmeticException e){

e-> we can extract the information from here

}catch(Exception e){

this can handle any thing and everything

}

System.out.println("Other Business Logic Goes Here");

if(condition){

if(condition){

}else{

}

}else{

if(condition){

}

}

try{

open connection (100 opened every second)

process with DB (error)

}catch(exception){

some message to exit gracefully

}finally{

try{

close connection

}catch(){}

finally{

}

}