Runtime :

Encapsulate Run-Time Environment.
Can't create object of this class but can get a reference to the current Runtime object by calling method "Runtime.getRuntime();"

↳ It controls state and behaviour of JVM.

methods

1. exec
2. freeMemory()
3. totalMemory()
4. gc()
5. static Runtime getRuntime()
6. runFinalization();

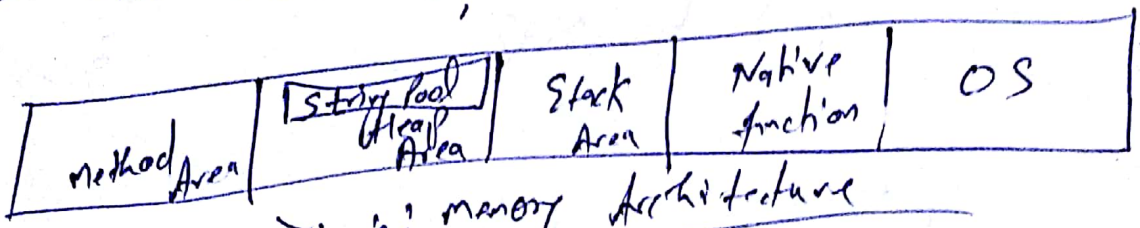

Diagram: memory Architecture

* Stack — Static Memory Allocation
eg: Thread, Variables local to a function in case of context switch

* Heap — Dynamic Memory Allocation

class A
{
private static A a = new A();
private A();
public static A getA()
}

Select TOP 1
from (select TOP 3 Salary
from Emp
Order by Salary desc) temp
Order by Salary

* Override hashCode() and equals() in case of User Defined Object as key.

int hashCode()
{
String str = name + surname;
return str;
}

27853930

boolean equals(Object o)
{
if (o == null)
return false;
Person p = (Person) o;
if (name.equals(p.name) && (surname.equals(p.surname))
return true;
}

(2)

```
class cfactorial
{
public: long fact;
        cfactorial (long fact)
        {
          this.fact = fact;
        {
        long factorial (long number)
        {
            if (number <0)
            {
              throw "Number can't be -ve";
            }
            if( number >= 1)
                return number * factorial (number - 1);
            else
                return 1;
        }
        getFact
```

SQL

```
-- 1. Equi Join
-- 2. Non Equi Join
-- 3.Self
-- 4.Inner
-- 5.Outer (Left, Right, Full)
-- 6. Cartesian

-- SubQuery
-- 1. Single Row Single Column
-- 2. Single Row Multiple Column

Select * from emp a
where 1 = (select count(distinct(sal)) from emp b
where b.sal >= a.sal)

Select  a.empno EmployeeNo,a.ename Employeename ,B.ename Manager  from
emp a, emp b
where a.Mgr = b.empno (+)
order by a.mgr desc

Select a.empno, b.deptno,b.dname from emp a, dept b
where a.Deptno=b.deptno(+)

--(+): means at NULL in Oracle  (Just opposite as in T-SQL /
ANSI)...There U concentrate on the table
--from which User needs complete records
--E.g:
Select a.empno, b.deptno,b.dname from
emp a
Left Outer Join dept b on
a.Deptno=b.deptno

--Right
Select a.empno, b.deptno,b.dname from
dept b
Right Outer Join
emp a on
a.Deptno=b.deptno

--Right for Dept No (Here will get Dept No. 40
Select a.empno, b.deptno,b.dname from
emp a
Right Outer Join
dept b on
a.Deptno=b.deptno


--FULL for Missing deptno'from DEPT and NULL For Unassigned DEPT NO. to
an EMployee from EMP
Select a.empno, b.deptno,b.dname from
emp a
Full Outer Join
dept b on
a.Deptno=b.deptno
```

```
--Ansii Equivalent of above Query (9i)
Select a.empno, b.deptno,b.dname
from emp a
Join
dept b
on a.Deptno=b.deptno
```

--Natural Join (Implicitly identifies relation between 2 tables (9i) feature. Here the column which relates 2 tables is not given any alias name (a.deptno……etc.)

```
Select a.ename, deptno from emp a Natural Join Dept b
```

```
-- SubQuery Operators: ALL, ANY
-- 1. Single Row Single Column
-- 2. Single Row Multiple Column
```

-- ALL: This row will display records of an employee whose Salary is Greater than all the salaries return in SUBQUERY

```
Select * from emp
where Sal >ALL (Select sal from emp where sal > 1000)
And
Job =Upper('Salesman')
```

-- Any: This row will display records of an employee whose Salary is Greater than all the salaries return in SUBQUERY

```
Select * from emp
where Sal >Any (Select sal from emp where sal > 1000)
And
Job =Upper('Salesman')
```

-- MULTIPLE COLUMN SUBQUERY (Here we are replacing AND operator after a where clause with one Subquery.

```
Select * from emp where ( sal, deptno )
in (Select max(sal),deptno from emp group by deptno  )
```

--Exist Operator is faster than "in" Operator as it will not traverse through all the records after finding the required matched record.

```
Select * from emp a
where exists (Select 1 from emp where a.deptno=emp.deptno and sal>4000)
```

--IN Operator is slower than "Exist" Operator as it will traverse through all the records even after finding the required match record.

```
select * from emp where
deptno in (Select deptno from emp where sal>4000)
```

Character Functions Returning Number Values

Character functions that return number values can take as their argument any character datatype.
The character functions that return number values are:

```
ASCII  INSTR  LENGTH
```

Datetime Functions

Datetime functions operate on values of the DATE datatype. All datetime functions return a datetime or interval value of DATE datatype, except the MONTHS_BETWEEN function, which returns a number. The datetime functions are:

```
ADD_MONTHS
CURRENT_DATE
CURRENT_TIMESTAMP
DBTIMEZONE
EXTRACT (datetime)
FROM_TZ
LAST_DAY
LOCALTIMESTAMP
MONTHS_BETWEEN
NEW_TIME
NEXT_DAY
NUMTODSINTERVAL
NUMTOYMINTERVAL
ROUND (date)
SESSIONTIMEZONE -- returns the value of the current session's time zone.
SYS_EXTRACT_UTC
SYSDATE
SYSTIMESTAMP
TO_DSINTERVAL -- INTERVAL DAY TO SECOND type.
TO_TIMESTAMP
TO_TIMESTAMP_TZ
TO_YMINTERVAL   -- INTERVAL YEAR TO MONTH type,
TRUNC (date)
TZ_OFFSET
```

```
class A
{
  int a;
  A() { a = 0; }
  A(int x) { a = x; }
  A(A x) { this.a = x.a; }
}

main()
{
  A a1 = new A(10);
  A a2 = new A(a1);
}
```

Super

Core Java (4/5)
Spring (3/5)
Hibernate (3/5)
SQL (3/5)
Servlets
C++
Linux / Shell

1, 2, 1, 3, 1, 2

1, 2, 3, 1, 2, 1

1, 2, 3, 1, 2, 1

1, 2, 3, 2, 1, 1

1, 2, 3, 1, 1, 2

1, 2, 2, 3, 1, 1

for(i=0; i<al..

1, 2, 3, 1, 2, 1 ✓

1, 2, 3, 2, 1, 1

1, 2, 3, 1, 1, 2

1, 2, 3, 1, 2, 1 ✓

al, i=0, j=i+1

1, 2, 3, 1, 2, 1

1, 2, 3, 2, 1, 1

SQL

Emp Id	name	emp manager id
100	ABC	101
101	XYZ	102
102	CBZ	

select empName
from Emp A, emp B
where A.empId = B.managerId

Hash map:

class employee
{
int id;
String fname;
String lname;
}

hash(code) equal() HashMap<Employee> hm = new HashMap<>();
hm.put(