DAY-14

------

OBJECT ORIENTATION

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

ORIENTATION --> view point.

OBJECT --> Real time entity.

RULES OF OBJECT ORIENTATION [FEATURES]

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

1. The world is a collection of objects.

2. Every object is useful, no obejct is useless.

3. Every object is in constant interaction with another object.

4. Every object belongs to a type, but type does not exist in reality only objects are exist.Object type is technically called as CLASS.

5. Every object will have two parts one is 'has' part also called as 'properties' and second one is 'does' part also calles as 'behaviour'.

6. Has part of a object is handled using 'datatype and variables' whereas does part is handled by 'methods'.

WHY WE NEED OBJECT ORIENTATION?

--> To solve real time problems we need object orientation.

example: BOOKMYSHOW application.

offline online

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

1.customer 1.customer

2.counter 2.counter

3.money 3.money

4.ticket 4.e-tickets

5.theater 5.theater

6.seat 6.seat

7.movie 7.movie

8.snacks 8.snacks

9.[black market] 9.not applicable

PILLARS OF OBJECT ORIENTATION:

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

1. CLASS

2. OBJECT

3. POLYMORPHISM

4. ENCAPSULATION

5. ABSTRACTION

6. INHERITANCE

1.CLASS

--------

--> Class is a blueprint/template which does not exist in reality.

SYNTAX OF THE CLASS:

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

access modifier Class Classname

{

variable

methods

blocks

constructors

interface

nested class

abstraction

}

SYNTAX OF A METHOD:

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

access modifier returnType methodName (parameter)

{

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

--------------- --> body

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

}

example for blueprint[class]

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

Class Lion

{

String name;

String breed;

String gender;

eat()

{

------

------

}

sleep()

{

-------

-------

}

run()

{

-------

-------

}

}

CREATION OF OBJECT:

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

An object can be created in 5 differnt ways:

1. using 'new' keywords

2. using newInstance()

3. using clone()

4. using FactoryMethod

5. using deserialization

CREATION OF OBJECT USING 'new' KEYWORD:

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

To create an object using 'new' keyword we must fallow 3 steps:

1. Declaration: declare a variable to a type(class) eg: Lion l1

2. Instantiation : allocate the memory (new) memory is pointed by reference variable eg: new Lion();

3. Initilization : variables and methods will be loaded to the memory.

ex: Lion l1 = new Lion();

example1:

---------

LionApp.java

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

class Lion

{

String name;

String breed;

String gender;

void eat()

{

System.out.println("lion always eats non-veg");

}

void sleep()

{

System.out.println("lion sleeps 10 hr per day...!");

}

void run()

{

System.out.println("lion runs faster than human being");

}

}

class LionApp

{

public static void main(String[] args)

{

Lion l1 = new Lion();

l1.eat();

l1.sleep();

l1.run();

}

}

output:

-------

lion always eats non-veg

lion sleeps 10 hr per day...!

lion runs faster than human being

NOTE: 1.Always program execution starts from the class where main method is present.

2.To access the variables and methods present in differnt class we will make use of reference varibale.

OBJECT VALUES CAN BE INITILIZED USING 3 WAYS:

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

1. using reference varibale

2. using methods

3. using the constructor

1. initilizing the object value using refernce variable

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

example

--------

LionApp.java

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

class Lion

{

String name;

String breed;

String gender;

void eat()

{

System.out.println("lion always eats non-veg");

}

void sleep()

{

System.out.println("lion sleeps 10 hr per day...!");

}

void run()

{

System.out.println("lion runs faster than human being");

}

}

class LionApp

{

public static void main(String[] args)

{

Lion l1 = new Lion(); // object creation is done using new keyword

l1.name = "simba"; // values for objects is given using ref variables

l1.breed = "Asiatic";

l1.gender = "male";

System.out.println(l1.name);

System.out.println(l1.breed);

System.out.println(l1.gender);

l1.eat(); // methods are called with the help of ref variable.

l1.sleep();

l1.run();

}

}

output:

-------

simba

Asiatic

male

lion always eats non-veg

lion sleeps 10 hr per day...!

lion runs faster than human being

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