Oops(Object Oriented Programming)//5-7------max(target)

1. Data Hiding

2. Abstraction

3. Encapsulation

4. Tightly Encapsulated Class

5. IS-A Relationship(Inheritance)

Multiple inheritance

Cyclic inheritance

6. HAS-A Relationship

Composition

Aggregation

7. Method Signature

8. Polymorphism

Overloading

♣ Automatic promotion in overloading o Overriding

♣ Rules for overriding

♣ Checked Vs Un-checked Exceptions

♣ Overriding with respect to static methods

♣ Overriding with respect to Var-arg methods

♣ Overriding with respect to variables

♣ Differences between overloading and overriding ?

Method Hiding

9. Static Control Flow

Static control flow parent to child relationship

Static block

10. Instance Control Flow

Instance control flow in Parent to Child relationship

11. Constructors

Constructor Vs instance block

Rules to write constructors

Default constructor

Prototype of default constructor

super() vs this()

Overloaded constructors

Recursive functions

12. Coupling

13. Cohesion

14. Object Type Casting

Compile time checking

Runtime checking

Difference between ArrayList l=new ArrayList() & List l=new ArrayList() ?

♣ In how many ways get an object in java ?

♣ Singleton classes

♣ Factory method

Data Hiding:

Our internal data should not go out directly that is outside person cant access our internal data directly

By using private modifier we can implement Data hiding

The main advantage of Data hiding is security

Abstract:-

hiding internal implementation of a class and exposing set of services is called abstarct:

Connection con=DriverManger.getConnection("databaseurl","username","password");

it has given me connection object

Encapsulation:

Binding data and member into a single unit is called encapsulation;

how to achive

declare member as private and expose through set/get;

Tightly Encapsulated class

A class is said to be tightly encapsulated if and only if every variable of that class declared as private whether the variable has getter and setter methods or not , and whether these methods declared as public or not, these checkings are not required to perform.

class Test{

private int x;

private int y;//valid

}

classs Test1 extends Test{

int x;//invalid

}