* Table user(uid, Uname, Pass, Fname, Lname, dob, dor);

int because it has more
operation than

Bean class

```
class user {
Private int uid, dob, dor;
Private string Uname, FName, Lname;
set --- ()
get --- ()
}
class DB {
addUser ( User obj ) {
    obj.get () ---
}
}
class MainApp {
    p s v m (s[] arg) {
    DB obj = new DB();
    User obj1 = new User();
    obj1. set --- ();
    obj. addUser (obj1);
    }
}
```

——— X ——— X ———

* Method Reference : calling another method from a method by reference.

① Non-static (access by object & anonymous object)
    method

```java
interface imr2 {
    void test();
}

class MR2 {
    public void show() {
        S.o.p(" method show");
    }
    public static void main(String[] args) {
        MR2 obj = new MR2();
        imr2 obj2 = obj::show;
        obj2 test();                    → by anonymus object

        imr2 obj3 = new MR2()::Show;
        obj3. test();
    }
}
```

② Static method (accen by class name)

```java
interface is1 {
    void test();
}
class MS1 {
    static void show() {
        S.o.p(" static show method"))
    }
    public static void main(String[] args) {
        imS1 obj1 = MS1::Show;
        obj1. test();
    }
}
```

③ Constructor Reference:

```
interface im3 {
    MR3 getMessage (String msg);
}

class MR3 {
    MR3(String msg){
        S.O.P(msg);
    }
}

class connf {
    public static void main(String[] args){
        im3 obj = MR3 :: new;
        obj.getMessage(" Hello constructor Refrce");
    }
}
```