**public** **void** setage(**int** age) {

**this**.age=age;

}

**public** **int** GetAge() {

**return** age;

}

**public** **void** setscores(**int** scores[]) {

**this**.scores=scores;

}

**public** **int**[] GetScores() {

**return** scores;

}

ATM {

Int accno;

Int passcode; // hide the data from external classes

}

HRDetails {

Private String Name; 1 + 1 ----- getHRName() , setHRName(param)

//Private Int hrempid; 1+1 --- getHRID() , setHRId()

//Private Double sal ; 1+1----getHRSal(),SetHRSal()

}

Class members are encapsulated within the class .

Simple variables

Int/string / double Getter () { // Access the data

Return val;

}

Public void Seetter(param ){ // To set the data for the class member

}

+++++++++++++++++++++++++++++++++++++++++++++

1. Declare private members
2. Define getters and setters

Set ---- 1

Get --- 2

**public** **class** EmployeeDetails

{

private int deptid[];

**public** **void** SetDeptID(**int** deptid[] ) {

**this**.deptid=deptid;

}

**public** **int**[] GetDeptID() {

**return** deptid;

}

}

//driver

**public** **class** AccessEmployeeDetails {

**public** **static** **void** main(String args[]) {

// create object for EmployeeDetails

EmployeeDetails ob = **new** EmployeeDetails();

**int** deptid[] = {1145,5643,7856,1980}; //values

//Array access

ob.SetDeptID(deptid); // taking values

**int** arr1[] = ob.GetDeptID();

//for each loop to traverse the array

**for**(**int** i:arr1) {

System.***out***.print(i + " ");

}

}

Abstraction ….

No detailed info available

Internal details must not be available

Skeletal system of the class

Methods without implementation

Empty framework without details

Abstract

* Abstract class
* Abstract methods --- no return type , return values
* Non abstract methods --- with implementation

To define abstract methods we need to create the derived class

Interface :

* Abstract methods
* To achieve multiple inheritance we shall use interface
* NO implementation of methods are allowed in side interface
* Keyword --- interface

Class A{

}

Class B{

}

Class C{

}

Class C extends A {

}

Class c extends B{

}

Interface

Interface A {

}

Interface B{

}

Interface C implements A, B{

}

Difference between Abstract class and interface ?

1. No method implementations are allowed inside the interface
2. We can achieve multiple inheritance in java using interface

Interface interfaceName{

Abstract void test1();

Abstract void test2();

}

Interface interfaceName2{

Abstract void test3();

Abstract void test4();

}

//default method is allowed interface