Last week coverage:-   int arr[];
* 1D-Arrays   arr = new int[10];
* Arrays (utils)
* ArrayList

Self-paced/later : 2D-Arrays, String, StringBuffer

Inheritance:-
- Extending attributes (data), behavior (operation)
   from one class to another class
- Resuability & Specialization
- is-a relationship

Examples:-
* Person
* Vehicle
* Shapes
* Account

Person	Student inherits Person	Person
fields:-	- urn	- Base class
- uid	- scores	- Super class
- fullName		- Generalized
- dob	+ constructors	
- gender	+ getter methods, additional	Student/Employee
	+ calculateGrade	- Derived class
methods:-	+ isPass	- Sub class
+ constructor(s)	------------------------------	- Specialized class
+ getter methods	Employee inherits Person	
+ setter methods, if needed	- empID	
+ computeAge()	- salary	
+ isMajor()	- designation	
	+ constructors	
	+ additional getter methods	
	+ payroll	
	+ SalaryIncrement	

```java
class A                                 class B extends A
{                                       {
        private int x;                          private int y;
        public A(int p)                         public B(int p, int q)
        {                                       {
                x = p;                                  super(p);    //will discuss more
        }                                               y = q;
        public int getX()                       }
        {                                       //TODO : getY
                return x;                       //TODO : printy
        }                                       //TODO: printxy
        public void printx()            }
        {
                //print x                main:-
        }                                B b1 = new B(10,20);
}                                        b1.printx();
                                         b1.printy();
                                         b1.printxy();
```

Lab : 1st October
===============

* Create an account on github.com
* Create a repo java-oop-labs
* Create 3 folders : Lab-5, Lab-6, Lab-7

Lab-5/StudentScoreAnalysis.java

Lab-6/SingleInheritanceDemo.java
Lab-6/MultilevelInheitanceDemo.java
Lab-6/HierarchialInheritanceDemo.java

Lab-7 ==> Polymorphism with Shapes
(Shape.java, Circle.java, Rectangle.java
Triangle.java, PolyDemo.java)

Lab-7

Poymorphism Demo - Shapes

Shape.java
Circle.java
Triangle.java
Rectangle.java
ShapeDemo.java ==> main

Shape
+ area
+ circumference

override them in respective class

```
Shape r1 = new Circle(10.5);          super class references holding
Shape r2 = new Triangle(3,4,5);       sub class objects
Shape r3 = new Rectangle(10,8);       - upcasting


r1.area();
r1.circumference();                   int ch;
                                      //read ch from keyboard

r2.area();
r2.circumference();                   Shape rs;
                                      if(ch==1)
                                            rs = new Cirlce(10.5); // rs = r1
                                      else if(ch==2)
r2.area();                                  rs = new Triangle(3,4,5); //rs=r2;
r2.circumference();                   else //ch==3
                                            rs = new Rectangle(10,8); //rs=r3;


                                      rs.area();
                                      rs.circumference();
```

Account class - super class

Data ==> id, name, balance

Methods ==>

+ constructor(s)

+ credit

+ debit

+ getBalance

SavingsAccount extends Account //balance

+ constructors

// override credit (deposit) & debit (withdraw)

CreditCardAccount extends Account //outstanding

+ constructor

// override credit (billpay) & debit (purchase)

github.com/caia-techblr/java-examples