Tuesday, April 26, 2022 9:49 AM

Alpha obj = new Alpha();

Alpha obj = new Beta();

Above equation is valid if and only if RHS isa subclass of LHS

Alpha obj = new Delta()	Valid
Beta obj = new Gama()	Valid
Theta obj = new Delta()	Invalid
Gama obj = new Theta()	Invalid
Beta obj = new Delta()	Valid
Alpha obj = new Theta()	Valid
Beta obj =new Alpha()	Invalid

HW1 ---- Call doJob 1 to doJob5 using all possible objects as discussed in class
After every call - run the program and see the output

Alpha obj = new Beta();

Reference obj has two data types !!!

1	The data type used at compile time	Static /compile time data type	LHS = Alpha
2	The data type used at run time	Dynamic/run time data type	RHS = Beta

		Compile time data type of obj	Run time data type of obj
Alpha obj = new Delta()	Valid	Alpha	Delta
Beta obj = new Gama()	Valid	Beta	Gama
Theta obj = new Delta()	Invalid		
Gama obj = new Theta()	Invalid		
Beta obj = new Delta()	Valid	Beta	Delta
Alpha obj = new Theta()	Valid	Alpha	Theta
Beta obj =new Alpha()	Invalid		

Object class is the super class of each and every class.

Type Casting -----
Upcasting Alpha obj = new Theta(); //sub class object is promoted to super class reference

Down Casting = super class reference is cast to a sub class type

```
Alpha obj ;

((Theta)obj ).fc();
```

The equals method of Object class ----------

Super class Object **implements the equals** method as follows ----

```
class Object
{
        public boolean equals(Object obj )
        {
                //the address of two objects is compared
                If ( this == obj )   return true;
                  else   return false;
        }
}
```

All classes have two options --

1. REUSE equals
   OR
2. OVERRIDE equals

---

super class implementation of equals is comparing address
But MyDate wants to compare content
SO MyDate should override equals !!!

INSTANCE = object

HW 2 ---- Complete the equals overriding of TechnicalBook , Patient and Employee

Test it in TestEquals class

HW3 --- fill the foll chart

	Alpha obj = new Beta()	Beta obj = new Theta()	Theta obj = new Gama()	Beta obj = new Delta()	Delta obj = new Delta()	Alpha obj = new Gama()
obj.fa()	Compile yes Output = Alpha					
obj.f1()						

obj.fb()	Compile no					
obj.fc()						
obj.fd()						
(Beta)obj).fb()	Compile yes Output = Beta					
(Theta)obj).fc()	Compile yes Output =crash					
(Delta)obj).fc()						
…						
…..						

Keyword = **abstract** !!!
abstract keyword is used for methods or classes NOT used for properties !!!

if a method is abstract = the method has no implementation

method with implementation are called as concrete methods

class that contains at least one abstract method MUST be abstract class
We cannot instantiate (create objects) of abstract classes
