Monsoon 2019

A Few More Topics

Object Oriented Programming

by

Dr. Rajendra Prasath

Indian Institute of Information Technology

Sri City – 517 646, Andhra Pradesh, India

Recap: Objects in JAVA ?

- An entity that has **state** and **behaviour** is known as an object
  - **Examples:** Chair, bike, marker, pen, table, car etc
  - It can be physical or logical
- An object has three characteristics:
  - **State:** represents data (value) of an object
  - **Behaviour:** represents the behaviour (functionality) of an object such as deposit, withdraw and so on
  - **Identity (Internally used):**
    - Signature (unique) of the object
    - Object identity is typically implemented via a unique ID
    - The value of the ID is not visible to the external user
    - But, Internally by JVM to identify each object uniquely

Recap: Collections

- **Collection:** Root interface with basic methods like add(), remove(), contains(), isEmpty(), etc
- **Set:** Doesn't allow duplicates. Example implementations of Set interface are HashSet (Hashing based) and TreeSet (balanced BST based).
  - Note that TreeSet implements SortedSet.
- **List:** Can contain duplicates and elements are ordered. Example implementations are LinkedList (linked list based) and ArrayList (dynamic array based)
- **Queue:** Typically order elements in FIFO order except exceptions like PriorityQueue.
- **Deque:** Elements can be inserted and removed at both ends. Allows both LIFO and FIFO.
- **Map:** Contains Key value pairs. Doesn't allow duplicates. Example implementation are HashMap and TreeMap.
  - TreeMap implements SortedMap.

- Difference between **Set** and **Map** interface: in Set, we have **only keys**, whereas in Map, we have **key, value pairs**

Instanceof Operator in JAVA

- The java instanceof operator is used to test whether the object is an instance of the specified type (class or subclass or interface)

- The instanceof in java is also known as type comparison operator because it compares the instance with type. It returns either true or false

- If we apply the instanceof operator with any variable that has null value, it returns false

Instanceof – An Example

- Let us look at an example:

```
class Shape { }

class Line extends Shape { }

class Square extends Line {

    public static void main(String args[]) {
        Shape s = new Shape();
        Line l = new Line();
        Square c = new Square();
        System.out.println(s instanceof Shape);
        System.out.println(l instanceof Shape);
        System.out.println(c instanceof Shape);
    }
}
```

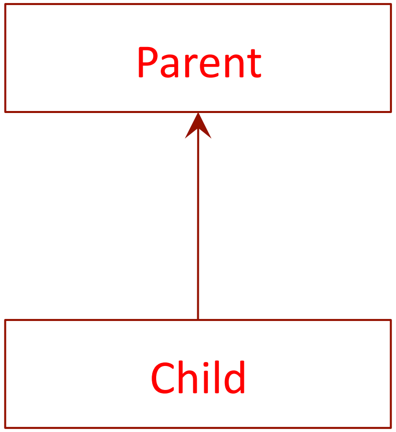
JAVA Typecasting

- Converting one type of value to another is called as Type Casting
- Different Forms of Type Casting:
- Two Types:
  - **Up Casting:**
    - Casting an instance of a child class to its parent class

      Example: **Shape s = new Line();**
  - **Down Casting:**
    - Casting an object of a parent class to its child class
    - If we perform it by typecasting, ClassCastException is thrown at runtime. But if we use instanceof operator, downcasting is possible.

      Example: **Square square = (Square) new Shape();**

Class Cast Issues



Parent

Child

✧ Let us look at the following:

Create an Object:

```
        Parent p = new Parent();
```

Upcasting:

```
        Parent p = new Child();
```

Downcasting:

```
        Child c = new Parent() ➔ Compile time error

        Child c = (Child) new Parent();
```

➔ This throws Class CastException at run time.

➔ How to solve this issue?

Downcasting (Instanceof)

- Example of Downcasting

```
class Parent { }

class Child extends Parent {
        static void doDowncast(Parent p) {
                if (p instanceof Child) {
                        Child c = (Child) p;  // Downcasting
                        System.out.println(“Downcasted!!”);
                }
        }
        public static void main(String args[]) {
                Parent p = new Child();
                Child.doDowncast(p);
        }
}
```

Exercise - 10

- **Create a plugin for Nutch Framework**
  - Hope that you guys have completed Nutch Code Walkthrough
  - Nutch has several plugins. They are several implementations of Multi-level and Hierarchical inheitance
  - Now consider the task: **Parsing of a webpage**
  - How do we achieve with your own parsing approach other than the one described in the nutch framework?

Assignments / Penalties

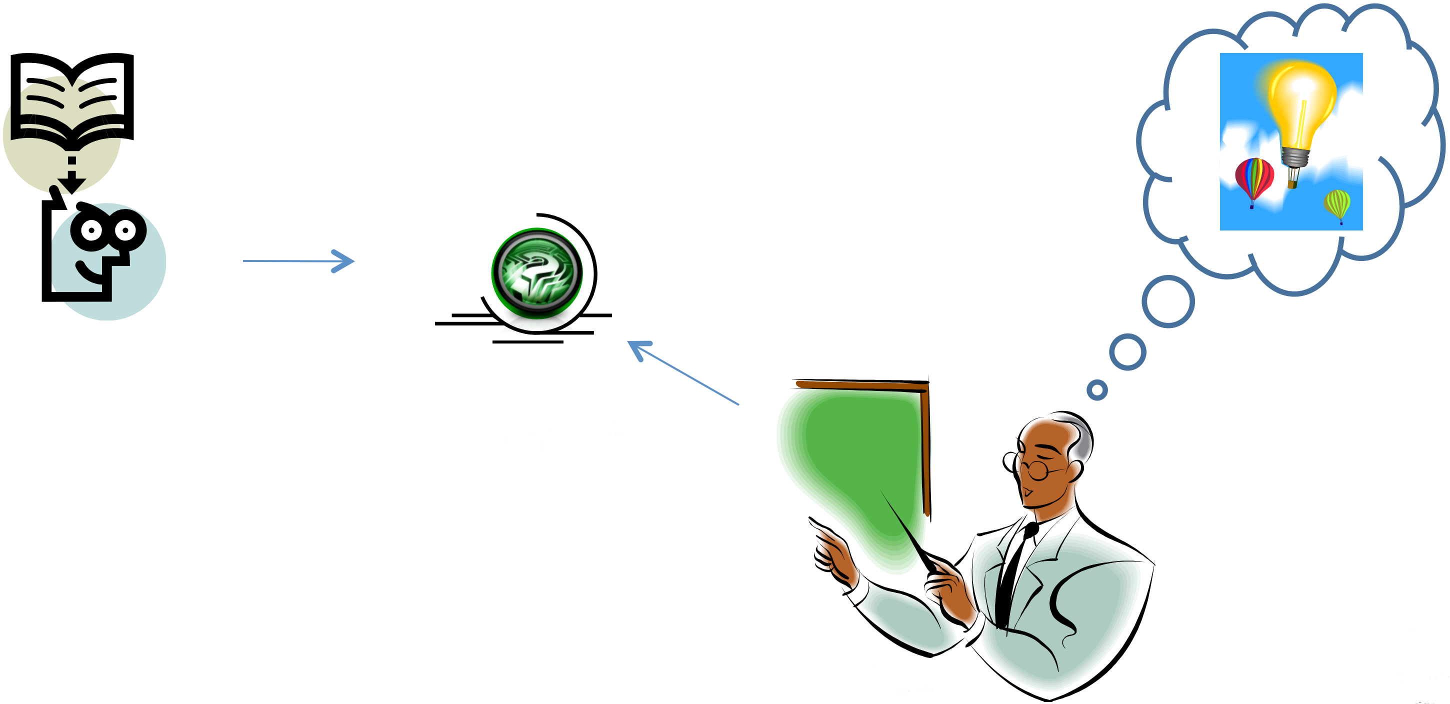
- ✧ Every Student is expected to complete the assignments and strictly follow a fair Academic Code of Conduct to avoid severe penalties

- ✧ Penalties would be heavy for those who involve in:
  - ✧ **Copy and Pasting** the code
  - ✧ **Plagiarism** (copied from your neighbor or friend – in this case, both will get "0" marks for that specific take home assignments)
  - ✧ If the candidate is **unable to explain his own solution**, it would be considered as a "copied case" !!
  - ✧ **Any other unfair means** of completing the assignments

Assistance

- ✧ You may post your questions to me at any time
- ✧ You may meet me in person on available time or with an appointment
- ✧ You may leave me an email any time
  (email is the best way to reach me faster)

Thanks …



… Questions ???