

# More on Abstraction in Java

## Deadly Diamond of Death

---

Produced      Mairead Meagher  
by:            Dr. Siobhán Drohan



Waterford Institute *of* Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics  
<http://www.wit.ie/>

# Deadly Diamond of Death!

---

- Recall that multiple inheritance is not allowed in Java.
- Any idea why the Java designers decided to not allow multiple inheritance?

# Deadly Diamond of Death!

---

- Recall that multiple inheritance is not allowed in Java.
- Any idea why the Java designers decided to not allow multiple inheritance?
- It is because of the Deadly Diamond of Death problem!

# Deadly Diamond of Death!

---

- This is easiest explained by example.
- Let's pretend that Java allows multiple inheritance and we will see really quickly what the Deadly Diamond of Death is!

# Deadly Diamond of Death - Example

---

- Suppose that we have an abstract super class, with an abstract method in it.

```
public abstract class AbstractSuperClass{  
    abstract void do();  
}
```

# Deadly Diamond of Death - Example

---

- Now two concrete classes extend this abstract super class.
- Each classes provides their own implementation of the abstract method defined in the super class.

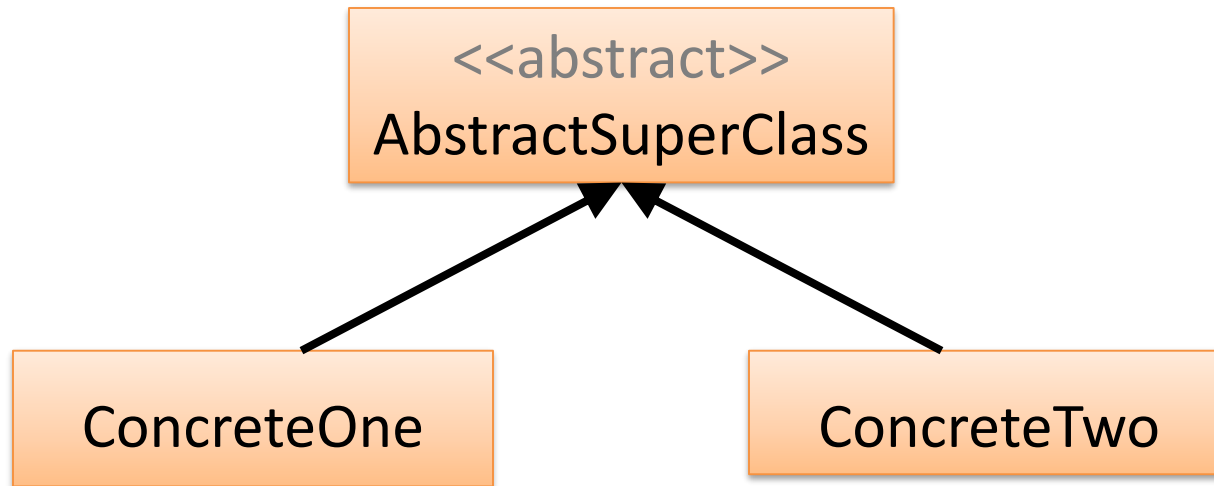
```
public class ConcreteOne extends AbstractSuperClass{  
    void do(){  
        System.out.println("I am testing multiple Inheritance");  
    }  
}
```

```
public class ConcreteTwo extends AbstractSuperClass{  
    void do(){  
        System.out.println("I will cause the Deadly Diamond of Death");  
    }  
}
```

# Deadly Diamond of Death - Example

---

- So far, our class diagram looks like this:



# Deadly Diamond of Death - Example

---

- Now a fourth class comes into picture which **extends** the above two concrete classes.

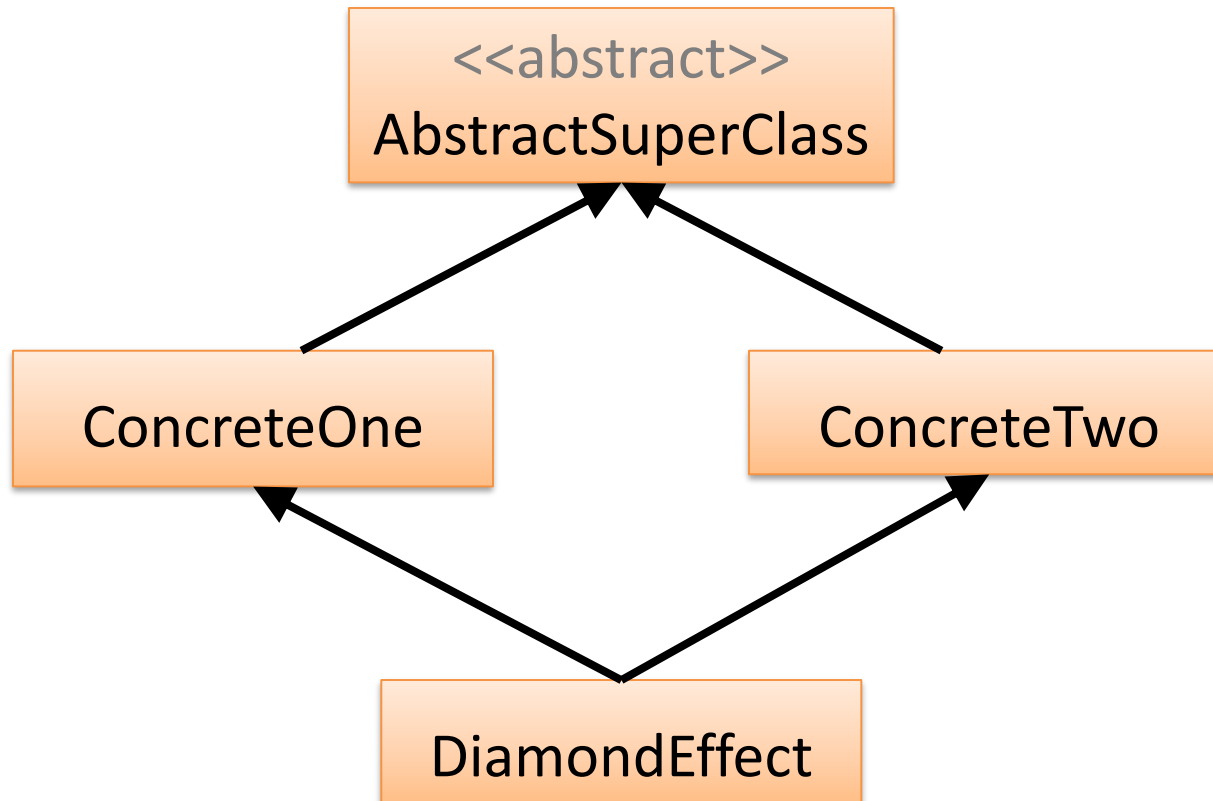
```
public class DiamondEffect extends ConcreteOne, ConcreteTwo{  
    //Some methods of this class  
}
```



# Deadly Diamond of Death - Example

---

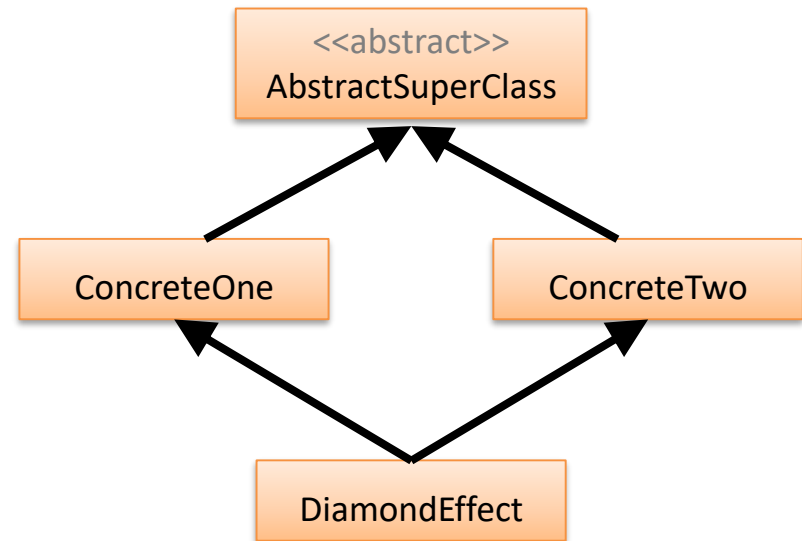
- Note that our class diagram is a diamond shape.



# Deadly Diamond of Death - Example

---

- The DiamondEffect class inherits all the methods of the parent classes.
- BUT we have a common method (void do()) in the two concrete classes, each with a different implementation.
- So which **void do()** implementation will be used for the DiamondEffect class as it inherits both these classes?



# Deadly Diamond of Death - Example

---

- Actually no one has got the answer to the above question, and so to avoid this sort of critical issue, **Java banned multiple inheritance.**
- The class diagram which is formed above is like that of a diamond, but with no solution or outcome, and so it is called Deadly Diamond of Death.

---

**Any  
Questions?**





Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>



Waterford Institute *of* Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics  
<http://www.wit.ie/>