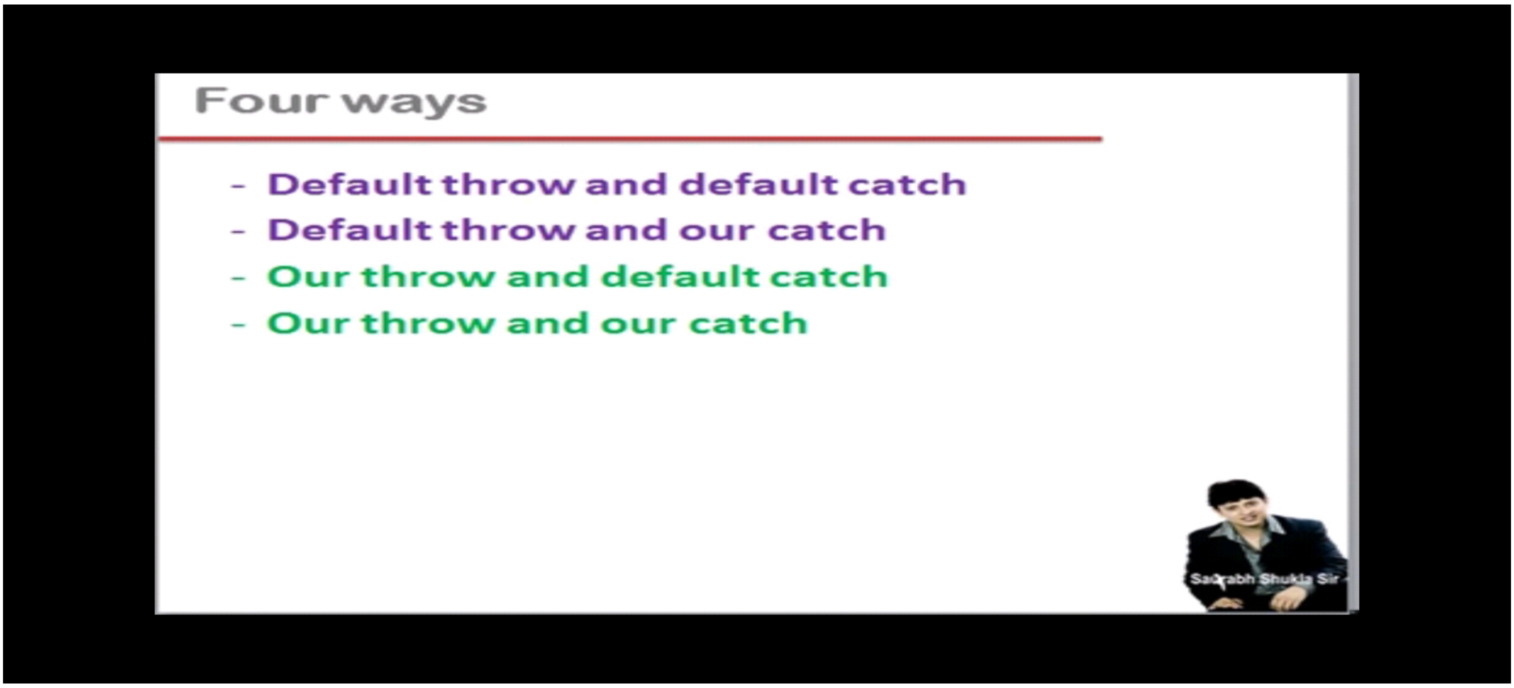
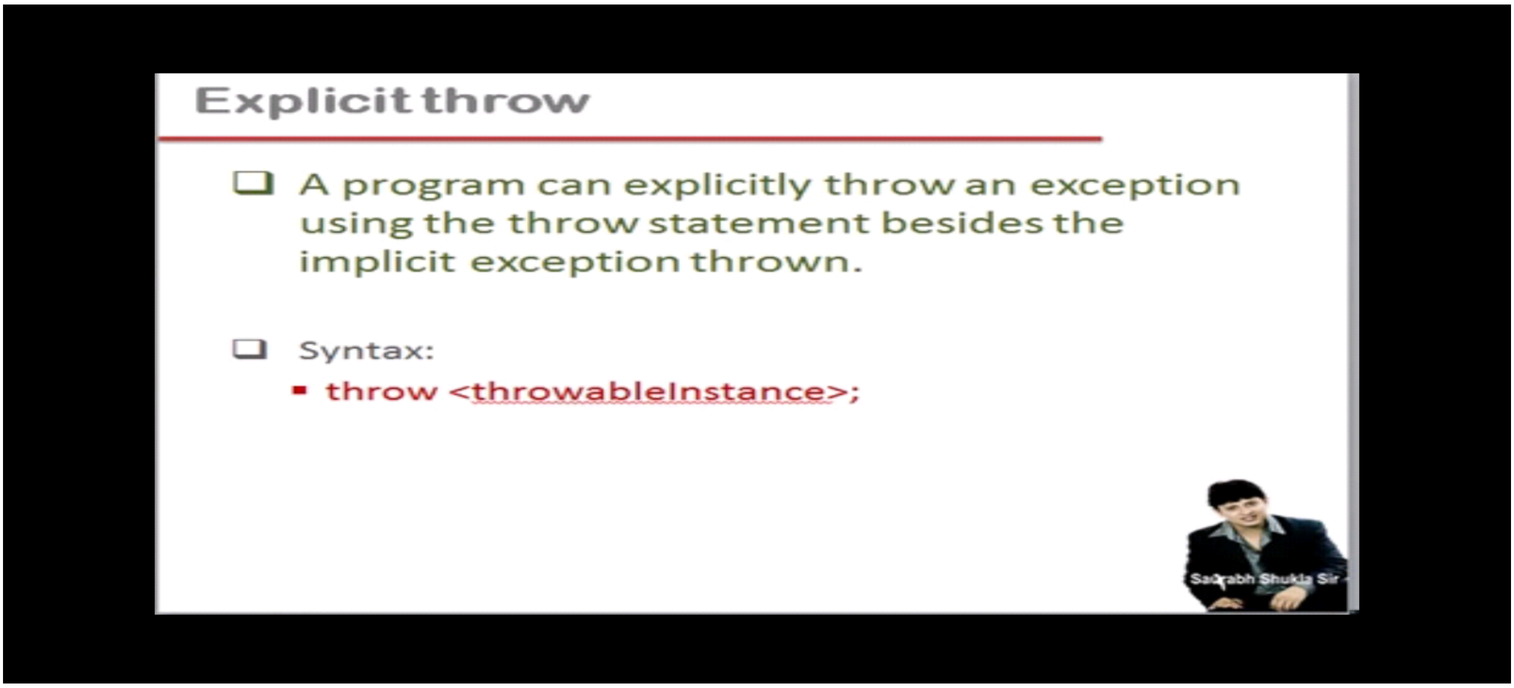


Four ways

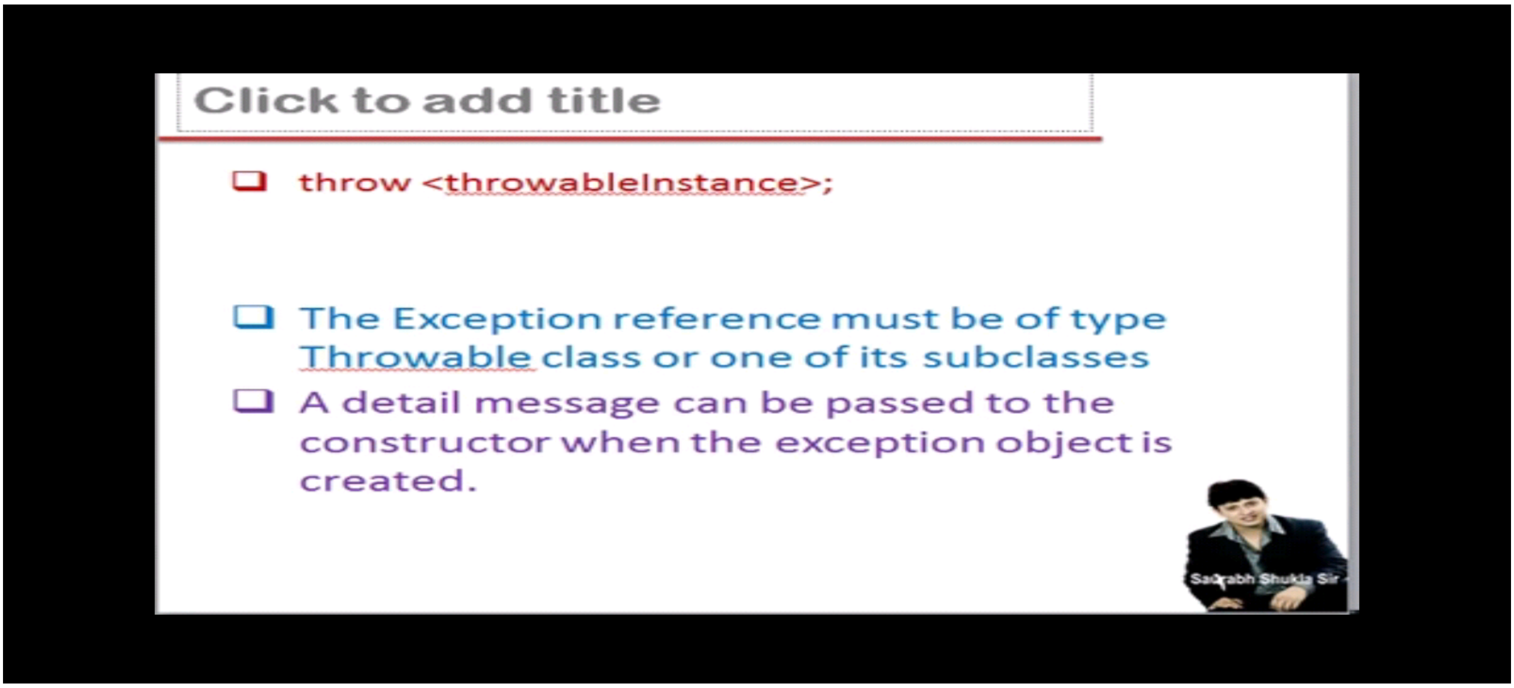
- Default throw and default catch
- Default throw and our catch
- Our throw and default catch
- Our throw and our catch



Explicit throw

- ❑ A program can explicitly throw an exception using the throw statement besides the implicit exception thrown.

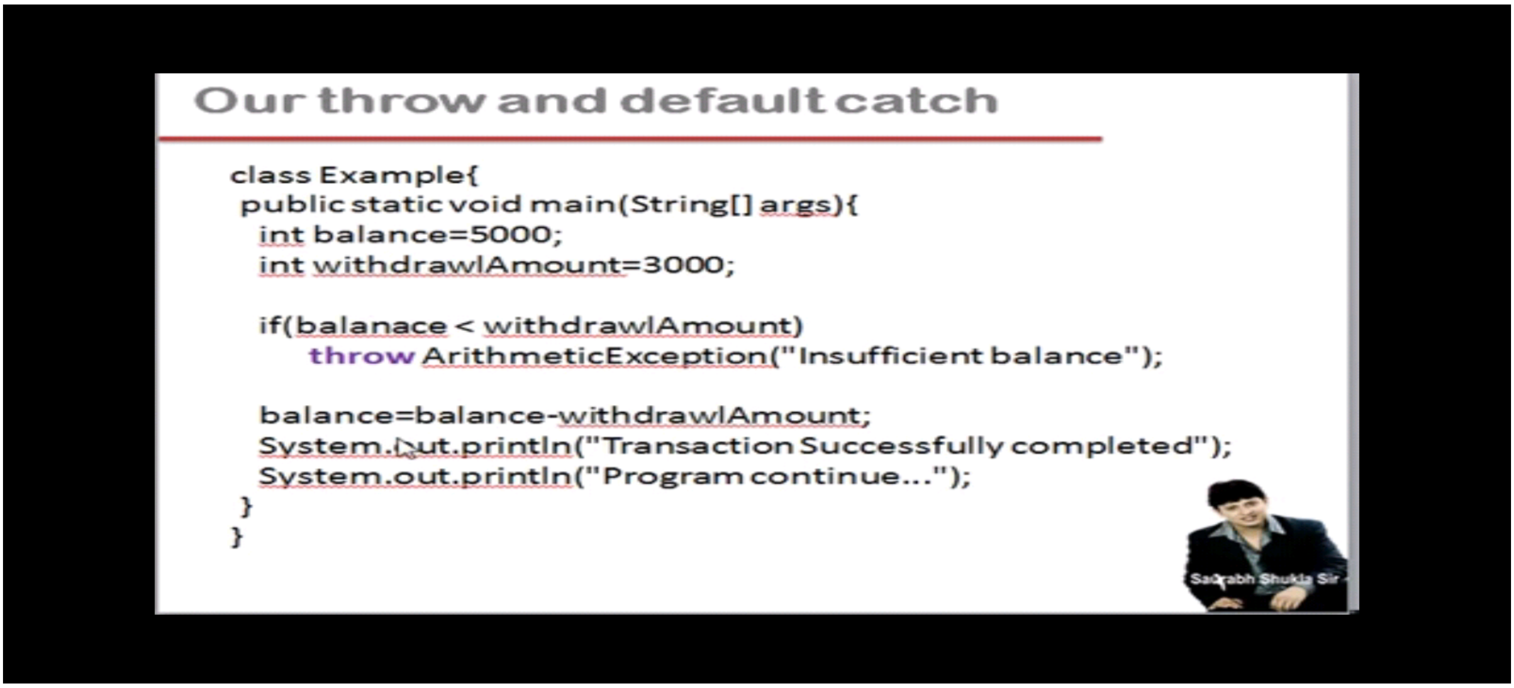
- ❑ Syntax:
  - throw <throwableInstance>;



Click to add title

- □ throw <throwableInstance>;

- □ The Exception reference must be of type Throwable class or one of its subclasses
- □ A detail message can be passed to the constructor when the exception object is created.

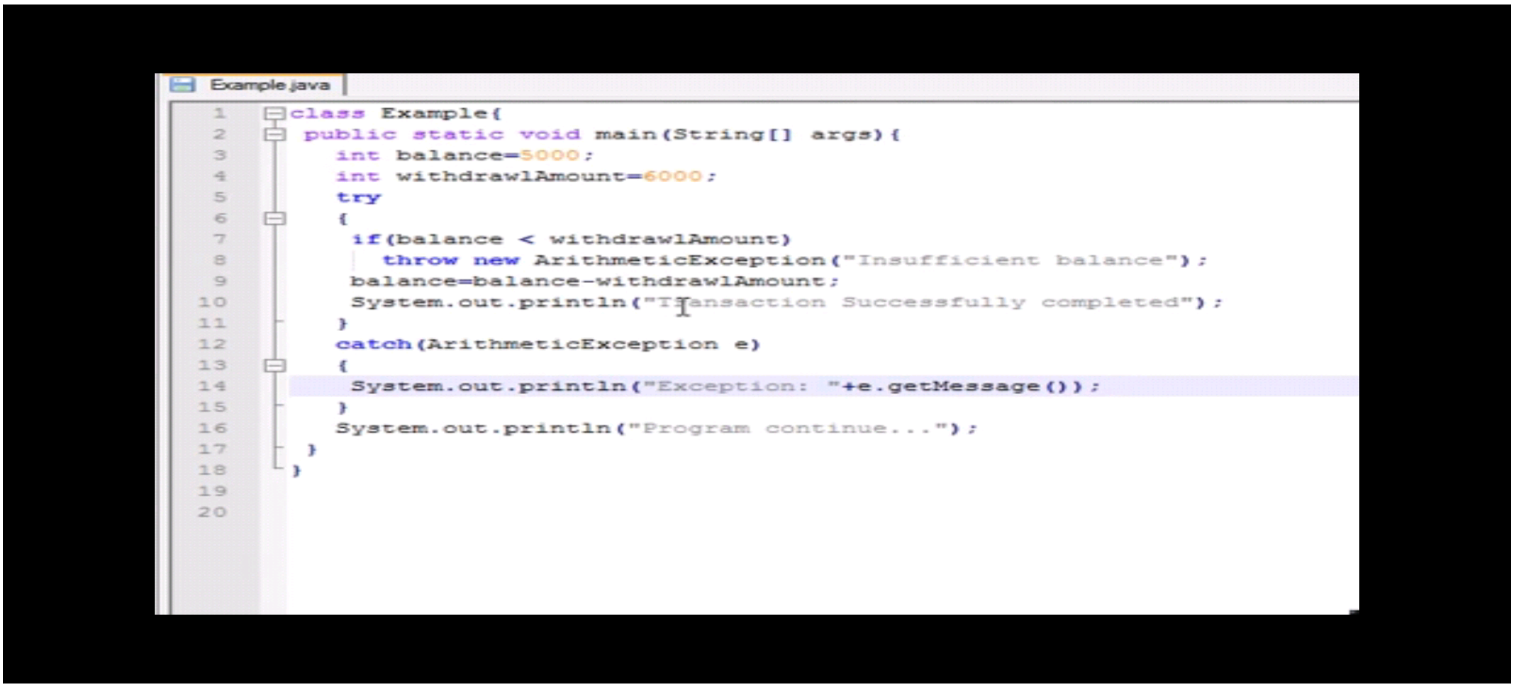


Our throw and default catch

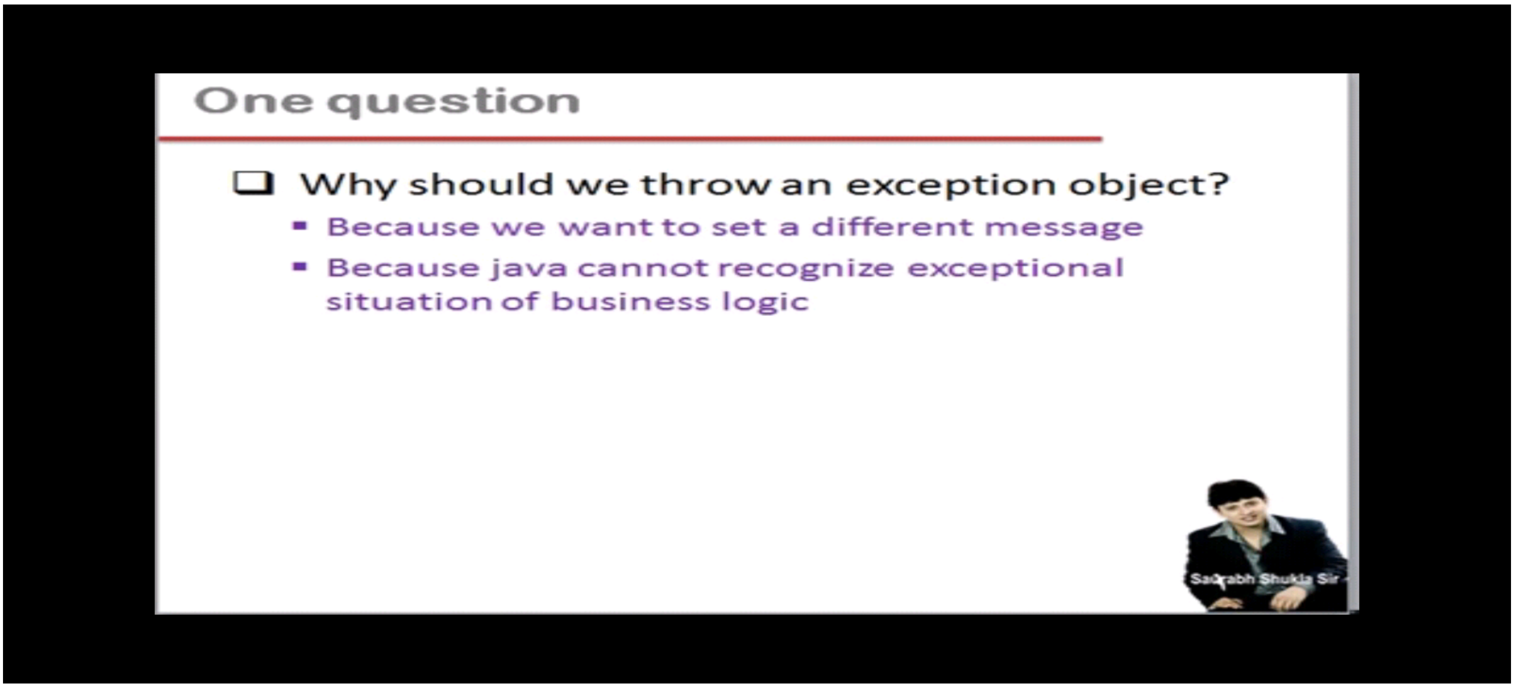
```
class Example{
 public static void main(String[] args){
   int balance=5000;
   int withdrawlAmount=3000;

   if(balanace < withdrawlAmount)
        throw ArithmeticException("Insufficient balance");

   balance=balance-withdrawlAmount;
   System.out.println("Transaction Successfully completed");
   System.out.println("Program continue...");
 }
}
```



```
class Example{
 public static void main(String[] args){
    int balance=5000;
    int withdrawlAmount=6000;
    try
    {
     if(balance < withdrawlAmount)
       throw new ArithmeticException("Insufficient balance");
     balance=balance-withdrawlAmount;
     System.out.println("Transaction Successfully completed");
    }
    catch(ArithmeticException e)
    {
     System.out.println("Exception: "+e.getMessage());
    }
    System.out.println("Program continue...");
  }
}
```



One question

- ❑ Why should we throw an exception object?
  - Because we want to set a different message
  - Because java cannot recognize exceptional situation of business logic