A try statement may have only one catch clause for each specific type of exception.

**try**

**{**

**number = Integer.parseInt(str);**

**}**

**catch (NumberFormatException e)**

**{**

**System.out.println("Bad number format.");**

**}**

**catch (NumberFormatException e) // ERROR!!!**

**{**

**System.out.println(str + " is not a number.");**

**}**



The NumberFormatException class is derived from the IllegalArgumentException class.

**try**

**{**

**number = Integer.parseInt(str);**

**}**

**catch (IllegalArgumentException e)**

**{**

**System.out.println("Bad number format.");**

**}**

**catch (NumberFormatException e) // ERROR!!!**

**{**

**System.out.println(str + " is not a number.");**

**}**

The compiler issues an error message regarding the second catch clause, reporting that NumberFormatException has already been caught. This is because the first catch clause, which catches IllegalArgumentException objects, will polymorphically catch NumberFormatException objects.

If you are handling multiple exceptions in the same try statement and some of the exception are related to each other through inheritance, then you should handle the more specialized exception classes before the more general exception classes.

**try**

**{**

**number = Integer.parseInt(str);**

**}**

**catch (NumberFormatException e)**

**{**

**System.out.println(str +**

**" is not a number.");**

**}**

**catch (IllegalArgumentException e) //OK**

**{**

**System.out.println("Bad number format.");**

**}**