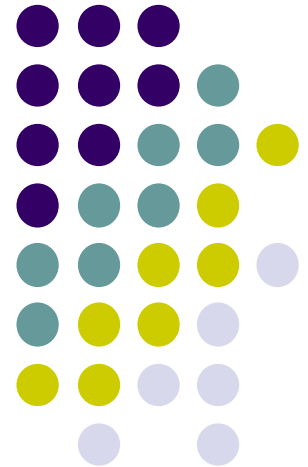


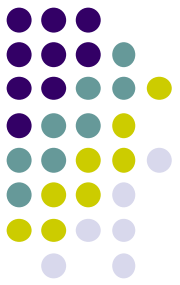
WIX1002

Fundamentals of Programming

Chapter 11

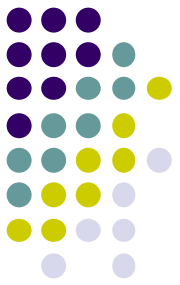
Exception Handling





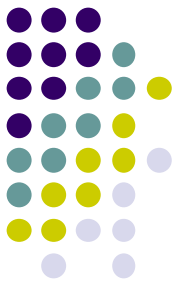
Contents

- Exception Handling
- try-catch
- Exception Class
- Exception in Method
- finally



Exception Handling

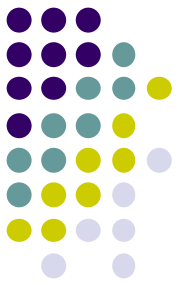
- **Exception handling** is a very important aspect of writing robust software. When an error occurs in a Java program it usually results in an exception being thrown.
- An exception represents an error condition that can occur during the normal course of program execution.
- When an exception occurs, an exception is thrown.
- By using exception handling, the exception is caught and processed.
- **try-catch** statement is used for exception handling.



try-catch

```
try {  
    // try block  
} catch (Exception e) {  
    // catch block  
}
```

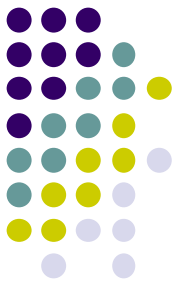
```
try {  
    throw new Exception("Exception Description");  
} catch (Exception e) {  
    System.out.println(e.getMessage());  
}
```



try-catch

- Multiple catch blocks

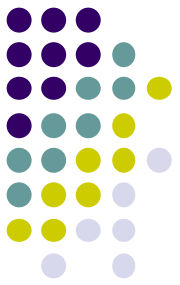
```
try {  
    // try block  
} catch (ExceptionOne e) {  
    // catch block  
} catch (ExceptionTwo e) {  
    // catch block  
}
```



try-catch

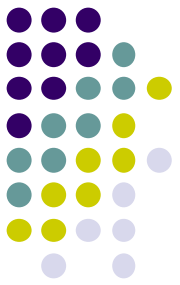
- Nested catch blocks

```
try {  
    // try block  
  
    try {  
        // try block  
    } catch (ExceptionOne e) {  
        // catch block  
    }  
  
} catch (ExceptionOne e) {  
    // catch block  
}
```



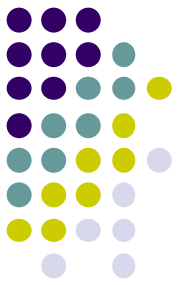
Exception Class

- Java includes some predefined exception classes.
- Some predefined exceptions are
 - IOException
 - NoSuchMethodException
 - FileNotFoundException
 - NumberFormatException
 - DivisionByZeroException
 - ArrayIndexOutOfBoundsException
- The new exception class can be defined. An exception class can be a derived class of any exception class.



Exception Class

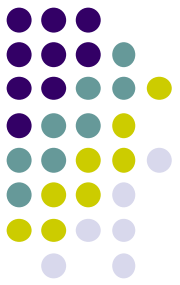
```
public class exceptionClassName extends Exception {  
    public exceptionClassName() {  
        super("Error Message");  
    }  
    public exceptionClassName(String s) {  
        super(s);  
    }  
}
```

Exception in Method

- Sometimes an exception can be thrown in a method without catching it in the same method.
- The method will stop if the exception is thrown.
 - `public returnType methodName(parameterType parameterName, ..) throws ExceptionName, ...`

```
try {  
    // try block  
    methodName();  
} catch (Exception e) {  
    // catch block  
}
```



finally

- The finally block contains code to be executed whether or not an exception is thrown in a try block.
- The finally always execute in the try block.

```
try {  
    // try block  
} catch (ExceptionOne e) {  
    // catch block  
} finally {  
    // Code to be executed whether or not an exception is  
    thrown  
}
```

