

JAVADOC DOCUMENTATION

CITS1001

Documentation

- Documentation is a **contract** between the **programmers** and the **users** of a class
- Documentation should be developed *at the same time*, and *in the same place* as the source code
- Java provides facilities to help with this, by allowing source code comments to be automatically turned into documentation using the javadoc feature

Javadoc

- Comments occurring between the special tags `/**` and `*/` are treated as special javadoc comments
- These can be automatically parsed (processed) to generate HTML documentation that can be read by a Web browser
- The resulting documentation is used to precisely define the behaviour of the class and its methods
- The Java API itself is generated from source code comments by javadoc
- But beware: only test cases can check whether the code actually does what the documentation promises

Writing Javadoc comments

- Place comments directly before a class, field (i.e. variable), constructor or method declaration
- Critically important for public classes, fields, constructors and methods, but may also be used for private ones if desired
- Comments can be written in HTML (the code used for Web pages), and enhanced with special javadoc tags such as

```
@author, @version,  
@param, @returns, @throws
```

```
/**
 * Create a new bank account
 * @param balance integer opening balance in cents
 * @param name String identifying the account and owner
 */
public BankAcc(String name, int balance) {
    this.balance = balance;
    this.name = name;
}
```

BankAcc(String name, int balance)

Create a new bank account

Parameters:

balance integer opening balance in cents

name String identifying the account and owner

```
/**
 * Creates and displays a SimpleCanvas of the specified size
 * with a white background. The client specifies whether repainting
 * after a drawing command should be manual or automatic.
 *
 * @param title title for the window
 * @param width the desired width of the SimpleCanvas
 * @param height the desired height of the SimpleCanvas
 * @param autoRepaint true for automatic repainting
 */
public SimpleCanvas(String title, int width, int height, boolean
autoRepaint) { //code here }
```

SimpleCanvas.SimpleCanvas(String title, int width, int height, boolean autoRepaint)

Creates and displays a SimpleCanvas of the specified size with a white background. The client specifies whether repainting after a drawing command should be manual or automatic.

Parameters:

title title for the window

width the desired width of the SimpleCanvas

height the desired height of the SimpleCanvas

autoRepaint true for automatic repainting