

# FIT1002 : Modularization

Gordon Ng

27 August, 2012 - 28 August,2012

# Contents

<b>1</b>	<b>UML</b>	<b>3</b>
1.1	Example . . . . .	3
<b>2</b>	<b>Encapsulation</b>	<b>3</b>
<b>3</b>	<b>Visibility</b>	<b>3</b>
<b>4</b>	<b>Our own classes</b>	<b>3</b>
<b>5</b>	<b>Instance Variable</b>	<b>3</b>

# 1 UML

- Upper Box: Name of Class
- Middle box: Attributes
- Lower box:
- + Public Class
- - Private Class
- method name, type

## 1.1 Example

+ test(): void

# 2 Encapsulation

It allows data and code that manipulates data to be visible or invisible from code outside the class.

# 3 Visibility

A instance variable or a method can be private which means only code inside the class.

There should be only one main method for each project.

# 4 Our own classes

There should be no main method and no static after public.

Static = I can do something.

# 5 Instance Variable

Their scope is the entire class they are declared in. Their lifetime is until they are destroyed by the garbage collector or any code.

## 6 Reminder

Rectangle r = create a reference variable that can point at any Rectangle2 object. It cannot point at anything other than a rectangle.

r = new rectangle(2,5) is

Initially it points at NULL. (This causes a NULL pointer error. )