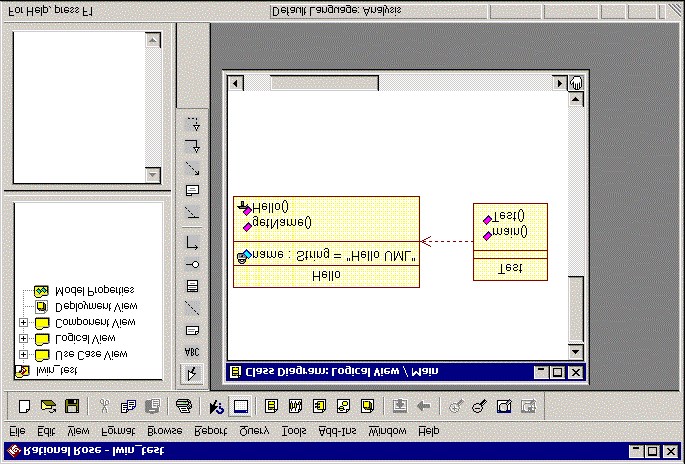
**EXPERIMENT 1: Date:**

**Aim:** Familiarization with rational rose

# PROCEDURE: A Tour of Rational Rose Environment

* Application Window: a title bar, tool bar, menu bar, and a work area for displaying toolbox, browser, document window, diagram window, and specification window
* Browser: a window with navigation tool for viewing diagrams and many other model elements
* Documentation Window: for preparing text information to describe model elements or relationships
* Diagram Windows
* Overview Window: can be showed through put the mouse pointer over hand icon
* Other windows such as Log, Edit, can be access through **View** on menu bar



Toolbox

Diagram

Window

Browser

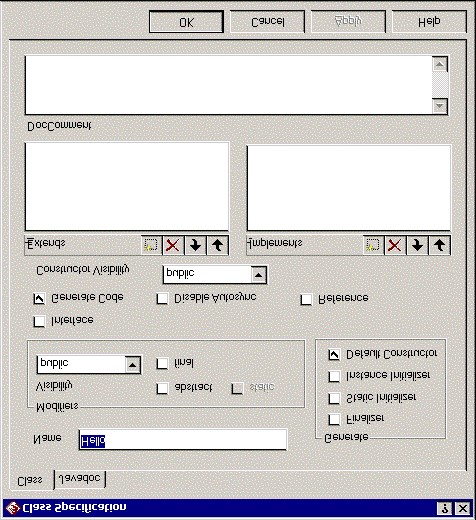
Documentation

Window

Title Bar

Menu Bar

Tool Bar



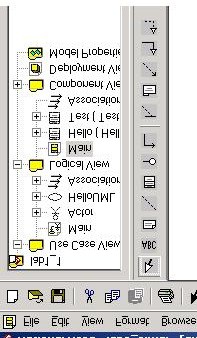
Specification

Window

**b) Creating a simple application (Hello.java) Step 1: Run Rational Rose Enterprise**

# Step 2: Create a Use Case Diagram

1. Click on Use **Case View > Main**

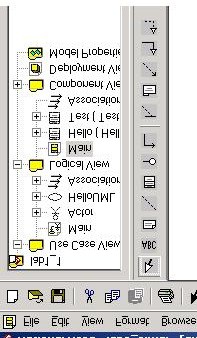


1. From the menu bar select **Tool > Create > Use Case**
2. Select the **Actor** from the Toolbox and place it into the Use Case Diagram
3. Right click on the Actor to see the pop-up menu, and then select **Open Specification** menu
4. Select the Use Case from the Toolbox and place it into the User Case Diagram
5. Right click on the User case to change the name to **HelloUML**

Actor HelloUML

**Step 3: Create Hello Class Diagram** (classes, attributes, and operations)

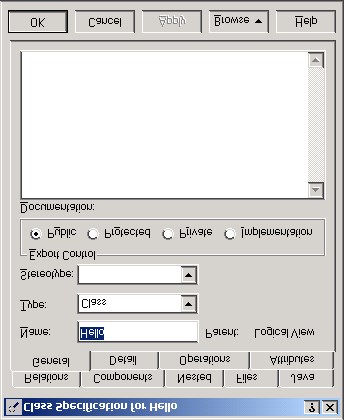
# Click on Use Logical View > Main



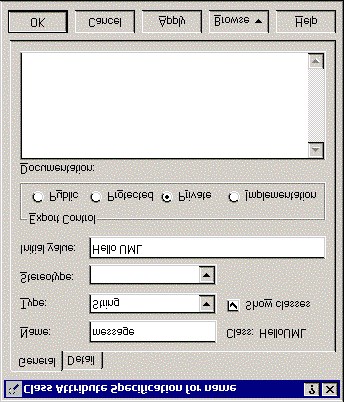
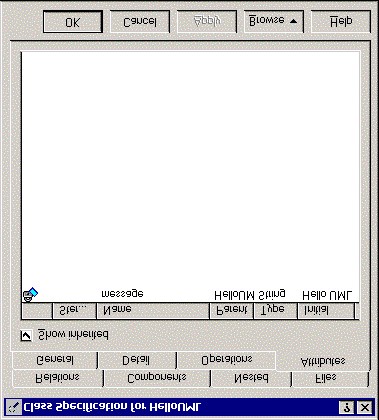
2. Create a new class called “Hello”

# - Tools > Create > Class

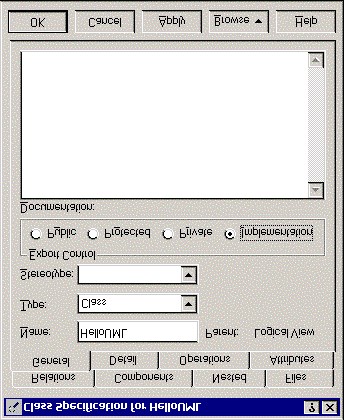
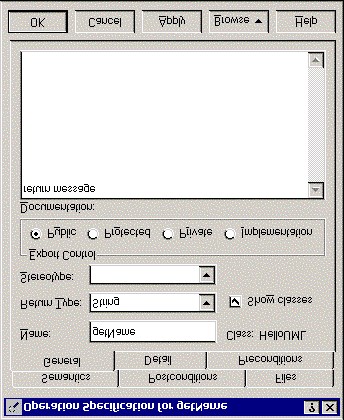
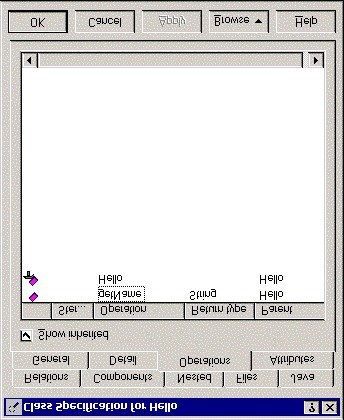
* Select the Class icon (rectangle box symbol) from the Toolbox
* Place a class to the workspace of the Class Diagram
* Type in a desired class name: “**Hello**”
* Right click the new class diagram and select “**Open Specification**”
* View the class name in the name box: **Hello** - Select “**class**” in the Type combo box



1. Add **Attributes** to the Hello class
   * Right click the Hello class to select **New Attributes** from a dialog box
   * Accept the default **name**, and hit enter
   * Right click the Hello class again to select **Open Specification** - The window called Class Specification for Hello is then displayed; change the attribute name to **message;** select Type as: **String**; specify Initial Value with a pair double quotes as: **“Hello UML”**; set Export Control to: **private**



1. Add **Operations** to the Hello class
   * Right click the Hello class to get an dialog box
   * Select **New Operations**, and give a new method name as **getName**, then hit Enter key to enter another method name as **Hello**
   * Double click the **getName** method to display Operation Specification menu for getName dialog box
   * Add return type as: **String**; add some information on this operation method to Document text box
2. Change Operation Specification for Hello method
   * Right click on the class diagram Hello to select Operation Specification for showing Class Specification for Hello
   * Select **Implementation** radio button in the Export Control
   * Verify the Hello class diagram is the same as shown below.



Hello



message : String = "Hello UML"



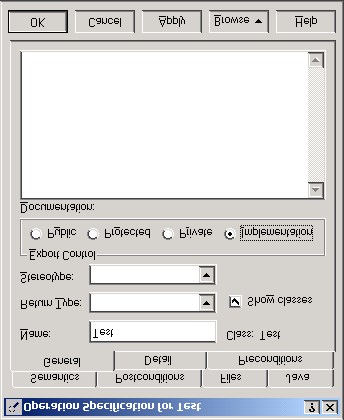
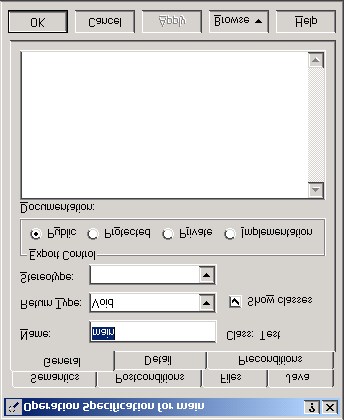
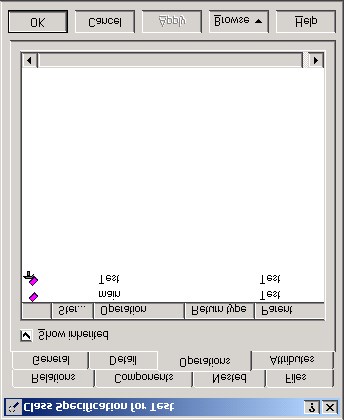
getName()



Hello()

# Step4. Create the Test Class Diagram

1. Select Class icon from the Toolbox
2. Place a class into the workspace Class Diagram, and give the class name: **Test**
3. Right click on the Test class, and select New Operations
4. Add two methods: **main** and **Test**



Test



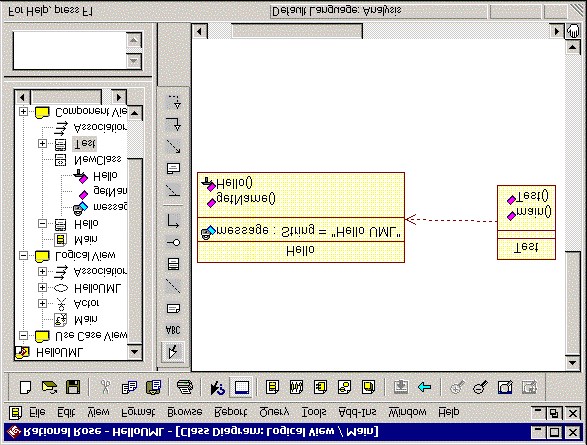
main()



Test()

# Step5. Object Instantiation

1. Select the dash line icon (Dependency or instantiates)
2. Draw the line starting from the Test to Hello to make an object instantiation



**Result:** The Design was successfully completed.