1. Getting Started

1.1 Launching BlueJ - UML

On Windows and MacOS, extracted the downloaded zip file and run the *BlueJ.exe* and *BlueJ.app* on Windows and MacOS correspondingly .

On Unix systems, installation is required. If you are using ubantu, installation could be perform by double click the downloaded BlueJ-3.1.4.deb. For others, Run the installer by executing the following command:

sudo dpkg -i bluej-3.1.4.deb

After installation, run the BlueJ deb package to launch the BlueJ UML

1.2 Explore Menus

The typical menus available on the menu bar of an BlueJ - UML window are:

- Project menu
- Edit menu
- Tool menu
- View menu
- Help menu

Menu Name	Description
Project	The Project menu allows you to open project for editing, save editor content and close the opened project so as the BlueJ UML. Among the other things, it also allows you to import existing BlueJ or BlueJ UML projects and create Jar Files.
Edit	The Edit menu allows you to perforfm new class, new package, add class from file, remove class, new uses arrow and inheritance arrow.
Tool	Compliation, rebuild package, reset Vm, use librart class and show use-case tool could be done using the Tool menu.
View	The View menu allows you to show dependency in class diagram, debugger, terminal and code pad.
Help	The Help menu can be used to search helper tips, add BlueJ extension.

2. Working with classes and dependency

2.1 Create BlueJ –UML Project

To create a project, select *New Project* from the Project menu. Select the project location and enter the Project Name. Click "Create" to crete the project.

2.2 Create class

Class could be created through selecting the "new class" menu item in edit menu tab, the "New Class..." tab on the left hand side and right click on the main window.

•••	BlueJ-UML (2.0.5): Test2
New Class	Car -y: Int -positiony: int + Car0 + SerPosition(position: int): void + forward(step: int): int New Class %N New Package %R

2.3 Remove Class

Class removal could be perform by right click the class and selecting the "Remove" menu item .



2.4 Compilation

To compile a class, click the Compile button in the editor. To compile a project, click the Compile button in the project window.

2.5 Edit soure code of a selected class

Double click the class name to open a editor and edit the code directly in the text editor. Save the changes by clicking "Save" in the "Class" Menu and compile the code.

• •	Save #S	Car - Test2		
ompile Un	Reload	Find Close	Source Code	
/** * Write * * Gauth	Page Setup Print #P Close #W ss or (your name)	: Can here.		
public c { // i priv	lass Car extends hi nstance variables - rep ate int position;	lace the example below with your own		
/** * C */ publ {	onstructor for objects ic Car() // initialise instance position = 0;	of class Car variables		,
} publ {	ic void setPosition(int this.position = positio	: position) m;		
publ {	<pre>ic int forward(int step // put your code here return position + step;</pre>	0		
3				
o changes ne	ed to be saved			read-or

2.6 Edit class attributes and methods

Double click the attributes/methods you would like to edit. The sign character '+' represents *public* publicity which could also be set to '-', '#', or '', respectively, for *private*, *protected*, or *package* visible sharing. Amendment on the parameter type and return type could also be made. After clicking the "OK" button and compilation, code will be updated instantly.



Changes could also made through correcting the code directly. Changes will be reflected in class after compilation.

2.7 Create linkage between classes

Class linkages including association, aggregation and composition could be created between classes. Association and aggregation relationship could be form by clicking the corresponding arrow button placed on the left hand side and a simple drag-anddrop operation as display in the follow image. Compile the program and the corresponding source code will be updated.



Linkage could also created through identifying them in source code. Compile the program and the constructed linkage will be reflected in class diagram.

2.8 Remove linkage between classes

Every linkage in a class diagram could be deleted by right click on it and click on remove button. Removal of any linkage in a class diagram will also reflect in source code. The original related source code would be updated to reflect the removal operation.



Same as linkage construction, linkage removal could be done by correcting source code. Corrsponding linkage will be destroy after compliation.

2.9 Editing multiplicities for aggregation and composition

BlueJ-UML shows multiplicity of aggregation and composition. Multiplicities could be modified by right click the aggregation/ composition and choose the "Change multiplicites" item. Source code will be corrected after compilation.



Multiplicites could also be ammened by changing the cource code directly. Changes will be reflected on the class diagram after compliation.



2.10 Edit role of dependency

Role of dependency could be updated by right click the linkage and select "Change Role". Modification will be reflected in souce code after compliation.



Similarly, updating role of dependency in source code will also update the class diagram after compliation.

3. Working with Objects

3.1 Create object instance

A key charactertics of BlueJ-UML is that you could directly interact with single class and execute their public methods instead of just execute a full application. Objet instance could be create by right click on any class and select "new ClassName()".

•••	BlueJ-UML (2.0.5): Test2
New Class	Car BlueJ: Create Object -w: Wheel[] // Constructor for objects of class Car -positiony: int // Constructor for objects of class Car +Car() +setPosition(position: int): void Name of Instance: car2 Ok
car1: Car	Wheel w firition int

3.2 execute methods of an object

After created an object, you can execute its public methds. Click with the right mouse button on the object and a menu with object operations will pop up The menu shows the methods available for this object and two special operations provided by the environment.



4. Use-case diagram



5. JUnit Test

5.1 Create Junit Test

To create a Junit test, click "new class" and select the type "Unit Test" like the following image.

BlueJ: Create New Class
Class Name:
Class Type
Class
Abstract Class
Interface
Applet
💿 Unit Test
◯ Enum
Cancel

The unit test class create imported the common Junit library like junit.Test. Common functions including setUp() and tearDown() are also included in the create unit test class created.



5.2 Create test method + Record test case

Test method could be created by right click the unit test and select "create test method" .



Another great function provided by BlueJ-UML is recording a test case. Test case record will be conducted after creating a test method. Executable code will be generated for every operation recorded.



Adding test case by writing code are also welcome in BlueJ-UML. Changes will be reflected in class diagram after compliation.

5.3 Run test cases

Click the "Run Tests" to run all the written test cases, or right click the unit test clas to select run a particular test case.

