**Importing a Project:**

A Menu used to import either a new project or an existing project based on your import source.it can be a java project or an android project, C/C++ so on.

**Exporting a Project:**

Once we finished a project we can export the project to a file. For e.g.: if we finish a project on android, we can create the apk of this project by exporting this project.

**Grouping:**

It is a way of representing your projects in a combined collection. You can set your custom collection of how it represents like a tree or another convenient structure.

**Project Creation:**

This is the first step to create a project. Steps are

File->New-> (Java project/Android Application project/Project)

**Working Sets:**

Working sets group elements for display in views or for operations on a set of elements.

The navigation views use working sets to restrict the set of resources that are displayed. If a working set is selected in the navigator, only resources, children of resources, and parents of resources contained in the working set are shown.

**Building and Cleaning:**

Builds work incrementally based on a previous built state. They will apply the transforms of the configured builders only on the resources that have changed since that previous state was computed (i.e., since the last build). Auto-building always uses incremental building for efficiency.

A clean build (**Project > Clean**) discards any existing built state. The next build after a clean will transform all resources according the domain rules of the configured builders.

Depending on your needs, build and clean can be done over a specific set of projects or the workspace as a whole. Specific files and folders cannot be built separately.

# Perspectives:

A perspective defines the initial set and layout of views in the Workbench window. One or more perspectives can exist in a single Workbench window.

Perspectives can be opened either in the same (existing) Workbench window, hiding the current perspective, or in a new Workbench window

Perspectives define visible action sets, which you can change to customize a perspective.  You can save a perspective that you build in this manner, making your own custom perspective that you can open again later.

# Views:

Views support editors and provide alternative presentations as well as ways to navigate the information in your Workbench.  For example, the Project Explorer and other navigation views display projects and other resources that you are working with.

Views also have their own menus. To open the menu for a view, click the icon at the left end of the view's title bar. Some views also have their own toolbars. The actions represented by buttons on view toolbars only affect the items within that view.

**Search Option:**

Text strings and files can be searched for in the Workbench. In this section, Searchwill be used to perform a text search across the resources that are shown in the navigation view. Instruction will also be given on how to use the Search view to work with the results.

**Compacting and organizing your code:**

A good programmer always writes their codes in a specific way. Better organizing the code makes easy to understand.

**Add:**

This Menu gives the option to add anything you want.it will be a class file, pictures, strings etc.

**Java naming Conventions:**

Java namingconvention is a rule to follow as you decide what to name your identifiers such as class, package, variable, constant, method etc.

But, it is not forced to follow. So, it is known as convention not rule.

All the classes, interfaces, packages, methods and fields of java programming language are given according to java naming convention.

Eg:

class Name: should start with uppercase letter and be a noun e.g. String, Color, Button, System, Thread etc.

interface Name: should start with uppercase letter and be an adjective e.g. Runnable, Remote, ActionListener etc.

method Name: should start with lowercase letter and be a verb e.g. actionPerformed(), main(), print(), println() etc.

variable Name: should start with lowercase letter e.g. firstName, orderNumber etc.

package Name: should be in lowercase letter e.g. java, lang, sql, util etc.

constants name: should be in uppercase letter. e.g. RED, YELLOW, MAX\_PRIORITY etc.