

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**  
**First Semester, 2016-17**  
**Object-Oriented Programming (CS F213)**  
**Lab -11**

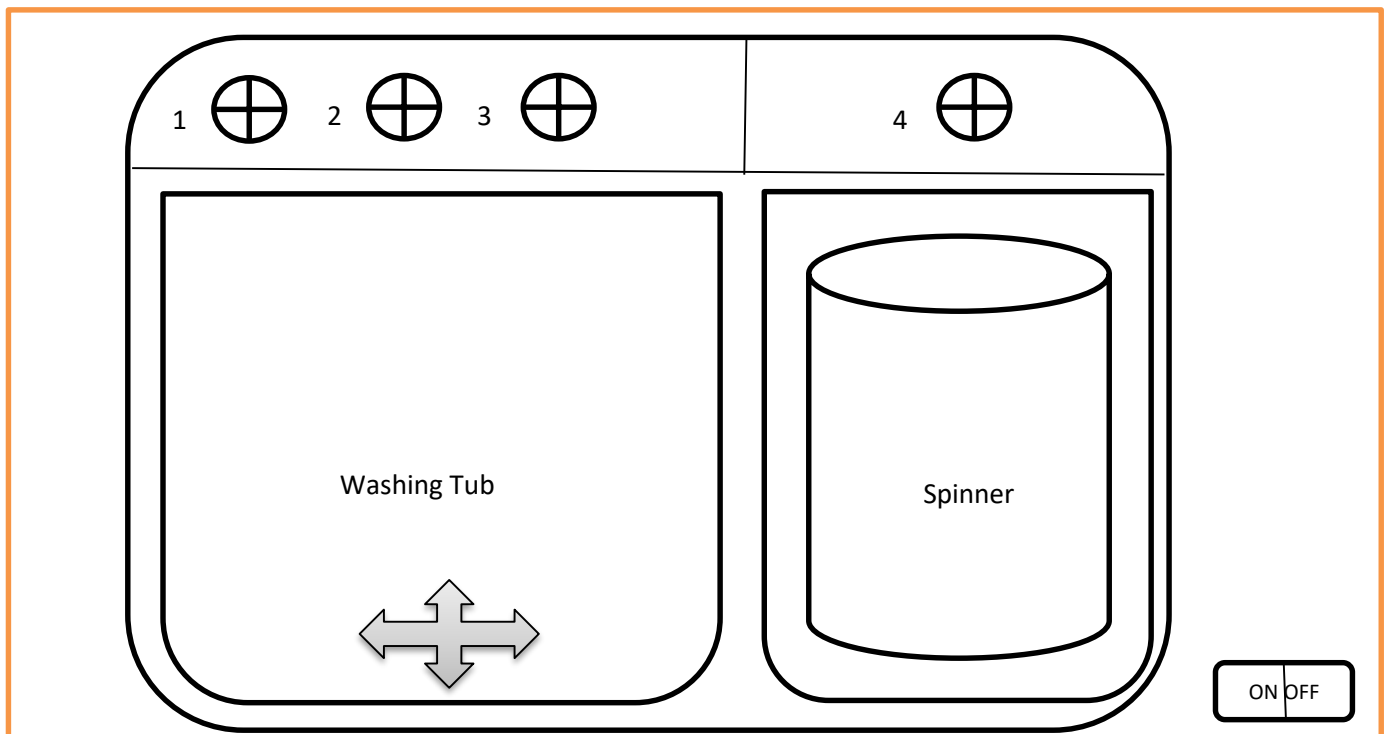
**Agenda**

1. Creating a washing machine simulator
2. GUI programming using Swing
3. Event handling
4. Exception Handling

**Overview**

In this lab, we will create a simulator of a washing machine using swing and event handling in java.

The purpose of this simulator is to showcase the working of the actual real world washing machine to the user. It will help the user get command over swing class components and understand how our life so much dependent on technology.



There are two containers of this washing machine simulator  
**1. Washer** for washing the clothes, and

## 2. Spinner for drying the clothes

Both these containers are controlled by several buttons provided for availing their functionality.

There are 5 buttons in the simulator:

**1. Power Button** – This button turns the machine on/off. Turning off this button will disable all the functions of the machine and turning On this button will make the machine operational.

**2. Water Inlet Spinner** – This component lets the user select the container in which the water has to be filled. The washer won't work until the water has reached the minimum level. Spinner chamber will work perfectly even without minimum level of water in washer chamber.

**3. Wash Timer Spinner** – This component allows the user to set for how long the clothes require washing and precondition for this button to be active is that power button must be turned on as well as minimum level of water must be present in washer chamber.

**4. Mode Spinner** – This button allows the user to select high, medium or low for the clothes that are put for washing.

**5. Spin Timer Spinner** -- This button allows the user to set for how long the clothes require drying and precondition for this button to be active is that power button must be turned on.

For implementing this we will use **Swing** class and **Event Handling** in java.

Now we will implement this scenario.

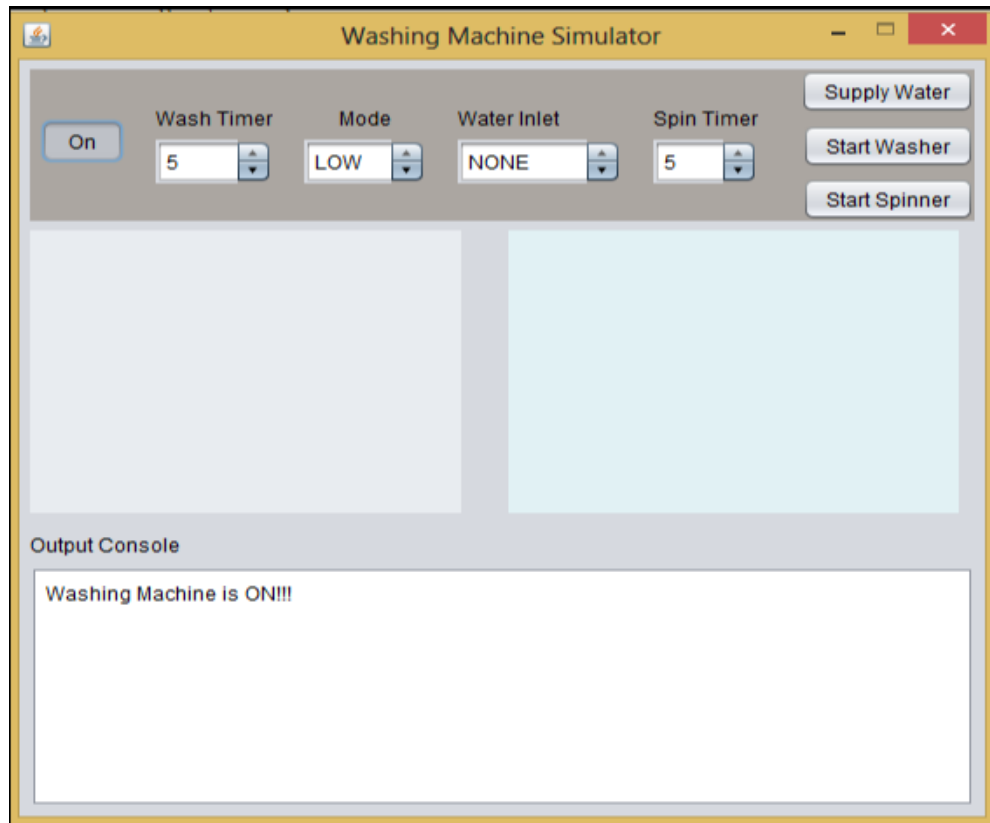
**A** Define these basic classes.

1. **WaterFiller** – This class keeps a check that before washing, minimum quantity of water is present and water doesn't exceed maximum quantity. This is implemented using threads which keeps track of water present in the washer.
2. **Timer** – This class handles washer time and spin time as set by the user. This is also implemented using threads.
3. **Container** – This class represents the container. This class holds the values for minimum, maximum and total capacity of the container. The specific containers ie Washer and Spinner have to extend this class and implement their own container specific functionalities.
4. **Washer** – This class defines all the attributes of the washer chamber and use getters and setters for these attributes.
5. **Spinner** -- This class defines all the attributes of the spinner chamber and use getters and setters for these attributes.
6. **Inlet** -- This class defines whether the mode set is washer, spinner or none. Use enum for defining this.
7. **Mode** -- This class defines the types of modes that the user can select. Use enum for defining this.
8. **ModeSelector** – This class contains a Mode member variable to keep track of the mode selected by the user.
9. **State** – This class defines the power button. Use enum for defining this.
10. **Status** -- This class defines the status of washer chamber whether the clothes are getting washed or water is filling or nothing is happening.
11. **WaterInlet** -- This class decided whether the water will be allowed to fill in the chamber by returning the state of water in the washer.

B. Steps for importing project in eclipse:

1. Go to File->Imports
2. Select Existing Project Into workspace
3. Select the browse button and select your project

B. Now implement **User Interface** for this simulator which looks like as shown in the figure.



It will include all the components described in the diagram above. Using swing class we will add each component to the frame and then set them the way we want them to appear when running the simulator. After that we will add event handling to these buttons so that when any button is pressed, it does the desired functionality.

Provide functionalities to the washing machine by completing the following methods:

```
private void onButtonItemStateChanged(java.awt.event.ItemEvent evt) {  
  
    /*This method is called from within the itemStateChanged method of ItemListener. This method  
    should enable all the components on the washing machine dashboard when the machine is  
    turned on. It should disable all the components on the washing machine dashboard when the  
    machine is turned off.  
    (Note that there are several threads that may be active when the machine is getting turned off.  
    Provide the functionality to kill those threads as well. See WMContoller Class for Thread Specific  
    code.) */  
  
}  
  
private void waterButtonItemStateChanged(java.awt.event.ItemEvent evt)  
{  
    /* This method is called from within the itemStateChanged method of ItemListener. This  
    method should turn on the WaterFiller thread to fill water inside the Washer container. The  
    water should be filled inside the container selected by the water inlet spinner. */  
  
}  
  
private void washerButtonItemStateChanged(java.awt.event.ItemEvent  
evt) {  
    /*This method is called from within the itemStateChanged method of ItemListener. This  
    method starts the washer. It should ensure that water is present inside the Washer container  
    atleast upto minimum level. The washer should run for a period of specified seconds as  
    specified by the Washer Timer Spinner. */  
  
}  
  
private void spinnerButtonItemStateChanged(java.awt.event.ItemEvent  
evt) {  
    /*This method is called from within the itemStateChanged method of ItemListener. This  
    method simply starts the Spinner for the specified amount of time by the user as specified by  
    the Spinner Time Spinner*/  
  
}
```