

# Coding Conventions

## Naming

- **Packages Name:** Starting with uppercase letter then followed by lowercase letters.

Example:  Controller  
 Model

- **Classes Name:** using UpperCamelCase.

Example: `class ExtractedTasks`

- **Methods Name:** using lowerCamelCase.

Example: `public boolean placeArmy(String country)`

- **Variables Name:** using lowerCamelCase.

Example: `private cardType cType;  
private Integer idCard;  
private Integer owner;`

## Code Layouts

- Methods have their opening braces on the same line separated by a space, and the ending brace of a block is in a full line alone.

Example: `public cardType getCardType() {  
 return cType;  
}`

- The braces are omitted for a control statement with only a single statement in its scope.

Example: `if(!checkValidityOfTasksList(tasksList))  
 return false;`

- No blank line between member variables in a class.

```
class CurrentPlayer{
```

Example: `private static CurrentPlayer currentPlayerObj = null;  
private ListIterator<Player> currentPlayerItr;  
private Player currentPlayer;`

- One blank line between methods in a classes.

Example:

```
public void setName(String name) {  
    this.name = name;  
}  
  
public Integer getNumberOfArmies() {  
    return numberOfArmies;  
}
```

- A single blank line between classes in a java file.

Example:

```
class ExtractedTasks {  
  
    public tasksEnum name;  
    public ArrayList<String> taskData;  
  
    public ExtractedTasks() {  
        taskData = new ArrayList<String>();  
    }  
}  
  
public class Controller {
```

- No blank line after comments that relate to methods or a block of code.

## Comments

- **File Header Comments:** a brief header block comment provides information helpful for understanding the program.
- **Single-Line Comments:** Use single-line comments to provide brief summary comments for chunks of code.
- **Documentation of Classes and Methods:** Providing JavaDoc comments for each classes and methods including @param and @return.