<http://www.journaldev.com/1763/template-method-design-pattern-in-java>

**Template Method** is a **behavioral design pattern** and it’s used to create a method stub and deferring some of the steps of implementation to the subclasses. **Template method** defines the steps to execute an algorithm and it can provide default implementation that might be common for all or some of the subclasses.

**package** com.journaldev.design.template;

**public** **abstract** **class** HouseTemplate {

//**template method, final so subclasses can't override**

**public** **final** **void** buildHouse(){

buildFoundation();

buildPillars();

buildWalls();

buildWindows();

System.out.println("House is built.");

}

//default implementation

**private** **void** buildWindows() {

System.out.println("Building Glass Windows");

}

//methods to be implemented by subclasses

**public** **abstract** **void** buildWalls();

**public** **abstract** **void** buildPillars();

**private** **void** buildFoundation() {

System.out.println("Building foundation with cement,iron rods and sand");

} }

}