# *Project 2: Shopping Application*

## Project Description:

**Note: Hand written documents not accepted. Should complete coding and the entire documentation in your laptops and showed when demanded. Hard copy to be submitted on 24th April 2017.**
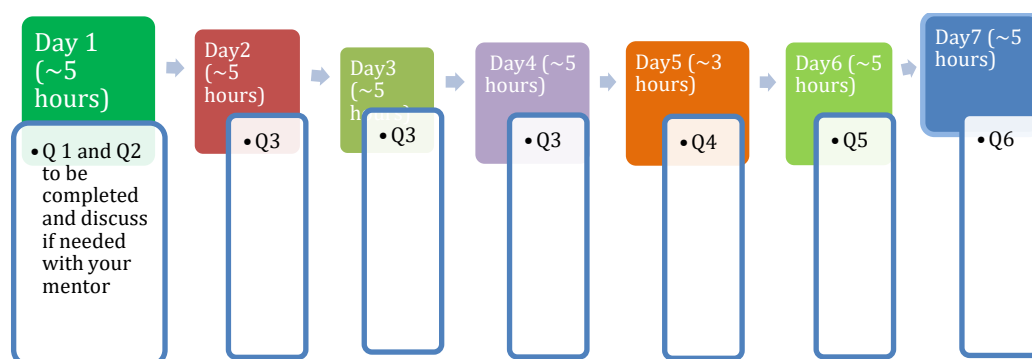
**Caution:**

**Mark allocation will be based on your understanding and if found guilty of copying will have to pay a huge penalty in terms of marks or failure.**

We have to create a online shopping software. The logged users are able to choose products from the catalogue and that product has to be added to the CART list. After adding all the products the user can be allowed to make payment for the product they chose. If the user is new he is allowed to create new account with all credentials (Descent information includes User name and Password).

If there's only one item available for each product, which means when any user adds this item to his shopping basket, the availability of that item will change to FALSE. The following details have to be maintained.

1. User List with contact details
2. Product List with number of items available.
3. Supplier List.

## *Your TimeLine*



| Day 1 (~5 hours) | Day2 (~5 hours) | Day3 (~5 hours) | Day4 (~5 hours) | Day5 (~3 hours) | Day6 (~5 hours) | Day7 (~5 hours) |
|---|---|---|---|---|---|---|
| • Q 1 and Q2 to be completed and discuss if needed with your mentor | • Q3 | • Q3 | • Q3 | • Q4 | • Q5 | • Q6 |

**Part 1:**

1. **Identify nouns and classes and make them BOLD. List the name of the classes, methods and variables**

| S.No | Class Name | Method Name | Variable Name |
|------|-----------|-------------|---------------|
| 1 | | | |
| 2 | | | |
| ……….. | | | |

2. **Draw the complete class Diagram with all the variables and methods including the constructors and Getters and Setters. Do indicate access specifiers, modifiers, static and constructions with accepted OOAD notations. Should explain in tabulation the justification for each of the OOAD concept used, the popular keywords (like has a, uses etc..) and the symbols. Note: Draw it manually using basic tools like paint etc.. And verify it with UML tools as many UML tools miss detailing.**

3. **Write Java codes for the classes you identified with appropriate comments and coding standards and generate Javadoc of your project. Note: More the scores, more you utilize the concepts you studied from unit 1 to unit 5.**

4. **Develop a Database of your choice and link it to your application.**

5. **Repeat step 4 replacing database with Files.**

6. **Develop a GUI using swings for the application you developed.**


**Part 2 (Optional):**

**Attempting this is Optional (Students who are explorative and good at time management, not recommended for moderate learner)**

7. **Apply Unit Testing for your application**