BACS2023 OBJECT-ORIENTED PROGRAMMING

ASSIGNMENT SPECIFICATION

Introduction	This is a group assignment. Students are to work in teams of maximum 3-4 members to create an object-oriented program for the given scenario.
Learning Outcomes Being Assessed	CLO2: Develop an object-oriented program using appropriate programming fundamentals with regards to arrays, methods and exception handling. CLO3: Implement the concepts of encapsulation, inheritance and polymorphism to solve a given programming problem.
Submission **Deadline**	**Week 12 (Friday) by 3:30pm** Late submission of the assignment will be handled according to **TAR UC's** Guideline for **Late Submission of Coursework**. No late assignments will be accepted (**get zero**). Please do not argue with your tutor if you really failed to submit your assignment on time as the consequence on late submission has been given in advance. However, in certain circumstances, the students may be allowed to turn in the assignment late. The students must contact the tutor BEFORE the assignment is due. The tutor will evaluate whether the circumstance warrants submitting the assignment late. A late penalty will be applied. The penalty is as follows: • Late submission within 1 - 3 days total marks to be deducted is 10 marks. • Late submission within 4 - 7 days: total marks to be deducted are 20 marks. • Late submission after 7 days: reject coursework and zero mark shall be awarded.
Overview	Ordering System

Description	In this assignment, students are required to build a professional **ordering system** with object-oriented programming (OOP) approach using Java programming language. You are required to develop simulated environment where output can be a console output of characters or graphical user interface (GUI). However, both display methods are acceptable without any marks discrimination. It is important to note that all students' team ideas must be **different** and **unique** from other student teams. You can innovate ordering system for different industry. E.g. food, groceries and etc. Your application is represented as Java objects in your system. You need to define the classes for these objects with advanced object-oriented programming features such as **polymorphism**, **inheritance** and **encapsulation** to control and manage the objects in your system. Also, you are encouraged to use **Java's interface** to establish weak relationship between objects in your system. You must provide reporting feature of your system for your users to monitor the status of your system. Do not use copyrighted images for your display if you are going for GUI display. You can create your own images or use public available licensed (e.g. creative common images) images with proper citation of the original source of the image.
Deliverables	The following items are to be handed in: (a) UML class diagram that depicts the entity classes and their relationships. (b) Cover page - Indicate the percentage contribution of each member. (c) Description of your team's assignment idea. Please provide sample screen shots and reports/listings. (d) Printing of source code is **NOT** required but you need to include your Java project source code in the softcopy submission. (e) Softcopy of source code – • Include **ALL** your source code files and all pre-compiled classes. • Form of submission: - Each group creates a folder named using the format TutorialGroup-StudentFullNamesWithAlphabeticalOrder(e.g.,RSF1(S2)-CheahLiMei-**H**engTzeSeong-**N**gSiewYongAlice), and to be attached together with the report. **IMPORTANT**: Work on the entity classes should be equally distributed between the team members. Likewise, for the client program and report/listing. **NOTE:** Submitting the assignment means you have agreed that your work is original and comply with the rules and regulations (refer to Academic Impropriety)

<table>
<tr><td><b>Paper Size / Submission Format</b></td><td>Digital Submission</td></tr>
<tr><td><b>Estimated Time Required</b></td><td>At least 10 hours per team member.</td></tr>
<tr><td><b>Academic Impropriety</b></td><td>You may only work with the students in your team to produce your deliverables for this assignment.
This covers cheating, attempts to cheat, plagiarism, collusion and any other attempts to gain an unfair advantage in assessment.
<b>The work that you submit must conform to those regulations.</b>
<b>NO-CHEATING POLICY</b>
A reminder on the no-cheating policy: You are <b>NOT</b> to share your work with your peers, but please feel free to have discussion with your peers. If cheating is discovered, both parties will take equal blame (<b>get zero</b>). Please note that the assignment should be your own work, although you may incorporate ideas or techniques from books, online resources, etc. By copying materials directly from any sources of materials will lead to zero. You have been warned. Whenever you face any problems, please seek advice from your tutor.</td></tr>
<tr><td><b>Assessment</b></td><td>This assignment contributes 80 marks to your coursework. The allocation of marks is shown below.
Refer to the <b><i>Assignment Feedback Form</i></b> for the detail assessment criteria.
<table>
<tr><th>Area</th><th>Marks Allotted</th></tr>
<tr><td>Completeness</td><td>15</td></tr>
<tr><td>Exception Handling</td><td>5</td></tr>
<tr><td>Implementation of Methods</td><td>10</td></tr>
<tr><td>Implementation of Arrays</td><td>10</td></tr>
<tr><td>Class Design</td><td>25</td></tr>
<tr><td>Class encapsulation and cohesion</td><td>35</td></tr>
<tr><td><b>Total marks</b></td><td><b>100</b></td></tr>
</table>
Marks for a team member = Total marks x % contribution</td></tr>
<tr><td colspan="2"><b>Note:</b> If it is an individual assignment, the total marks will not be multiplied by the % contribution.</td></tr>
</table>