GEORGE MASON UNIVERSITY | IT 206
Object-Oriented Techniques for IT Problem Solving

Programming Assignment 5/6

Scenario:

Your work at Sew What? Gifts and Knickknacks continues to be praised. Monica was so pleased with your work on her payroll application, now she wants your help to create an application to run the company's daycare program, which is currently in a startup phase.

To get off the ground running, Monica is restricting children that can join the daycare to those who are between 3 and 11 years of age, inclusive of these ages. Monica needs to be able to track all of the children who join the daycare. However, since the program is still in its startup phase, the program can only support up to 8 children, with the understanding this number needs to be able to be changed in the future.

For each child, the application must store the child's name, age, street address, phone number, email address, and a list of food allergies. To be eligible for this daycare program, children may not have more than 7 food allergies. A valid phone number is in the format: (xxx) xxx-xxxx. An example phone number is: (703) 993-9999. A valid email address is in the format: xxx@yyy.com, whereby at least one character must be placed before and after the at sign (@). The number of letters after the period must be 2, 3, or 4. An example email address is: bsit@gmu.edu.

Children are registered for the daycare program for up to 5 days per week. Some children require care with extended hours. Parents are charged a fee of $106.26 per day registered for the daycare program. An additional $16.96 surcharge, per day, is added for children who require care with extended hours. For children who require care with extended hours, it must be held across all days the children are registered for the daycare program. It is not possible to have care with extended hours on some days, but not other days. For example, a parent with a child who is registered for the daycare program for 3 days per week and requires care with extended hours will pay $369.05

Some children also receive tutoring services. Your application must have a way to create a tutored child. A tutored child contains all of the same information as a regular child does, but in addition, you must also track the child's grade level and the number of subjects he/she will be tutored in. Parents must pay a surcharge of $12 per subject on top of all other costs. For example, a tutored child who is registered for the daycare program for 3 days per week and receives tutoring in 3 subjects will pay $354.78.

Create an efficient, object-oriented solution with a polished user interface that will allow Monica to manage the company's daycare program. Monica must be allowed to enter in each child's information (in full), one at a time, until she has indicated she is finished entering children, or the daycare program fills, whichever comes first. If Monica enters a child that is ineligible to be added to the daycare program, she must be informed of this, and the child's information should be discarded. She can then proceed entering other children's information.

Upon completion of entering child information, Monica must see a list of all children and all information associated to each child in a well-formatted report. The report must also include the total number of children who are tutored, total number of children who are not tutored, average cost of children who are tutored, and average cost of children who are not tutored.

Other Requirements:

- Your solution must use object-oriented techniques (No points earned for a procedural solution).
- Your solution must not import any Java library other than JOptionPane.
- Your solution must demonstrate use of arrays.
- Your solution must demonstrate the concept of inheritance.
- Your solution must be designed as a modular solution using methods other than main, with each method performing one task.
- Your solution must include appropriate constants, constructors, accessors (including a toString() method), validating mutators, and special purpose methods
- You may not use Scanner or System classes for input/output. You must use JOptionPane.
- You may not use System.exit, or any variant that exits the program in the middle of the program. The program should only exit once the algorithm has finished completing.
- Your solution may not use any functions or language constructs not covered during IT 106 or this semester's IT 206 without prior authorization from your instructor, even if you know other functions or language constructs. We want everyone to be on the same "playing field", regardless of previous programming exposure, and get practice with algorithmic design to solve problems (the intent of the course). Using something existing not discussed in class does not give you as much practice as solving the problem yourself. **Doing this may lead to a substantial grade penalty, a grade of zero, or an Honor Code inquiry.** When in doubt, ask!

To Do (Check Blackboard for Due Dates):

Programming Assignment 5: Solution Design

1) List and describe the purpose of each class that will be needed to solve this problem
    a. You must separately identify (list) and describe the purpose of each class. One or two sentences per class should be sufficient.
2) *Data Definition Class(es)* - Create a detailed UML class diagram, listing and explaining all class variables, accessors, mutators, special purpose methods, and constructors associated to each data definition class.
3) *Implementation Class* - Create a table that lists all methods that will be used to create the implementation class. For each method identified, provide the following:
    a. A 1-2 sentence describing the purpose of the method

b. A list of the names, data types, and brief description (1-2 sentences) for each input variable into the method, if there are any

c. The name and data type of the variable to be returned from the method, or void if nothing will be returned

Note: You do not need to provide any pseudocode. However, you should create a plan for yourself as to how you will perform the logic for each method. If you do not do this, you will have great difficulty in completing the solution implementation.

An example format to use for this table is as follows. You can format this table in any way you'd like, so long as the information is clearly presented. Completing this table correctly will help you build your documentation and code for your solution implementation.

Method:	getMagazine
Purpose:	The method will allow for the creation and population of a Magazine object based on user input of a title, cost, and number in stock
Inputs:	none
Return:	magazine : Magazine – The Magazine object created and populated from user input

Method:	checkout
Purpose:	The method will allow for the purchase of a magazine. It takes into account the sales tax that will be charged as part of the cost calculation
Inputs:	magazine : Magazine – The magazine to be purchased salesTax : double – The sales tax percentage to be charged as part of the checkout
Return:	void

Upload a Word document containing only items above to Blackboard.

Grading Criteria	
Requirement	**Points**
List and describe the class(es) needed to solve the problem	10
Data Definition Class Design – Detailed UML Diagram	40
Implementation Class Design – Table Listing All Methods	50

Programming Assignment 6: Solution Implementation

Write a well-documented, efficient Java program that implements the solution design you identified. Include appropriate documentation as identified in the documentation expectations document.

To Blackboard, **submit ONLY ONE .zip file** containing all of the .java files part of your submission for your solution implementation. Your .zip file should contain only the .java files in your solution. Be careful that you do not submit .class files instead of .java files.

Warning! You must submit **ONLY** **ONE** .zip file containing **ONLY** your .java files. Failure to follow this instruction precisely will result in a 10 point deduction of the assignment score. **No exceptions!**
Why is this important? The goal is to teach you how to properly package your IT solutions into a "customer-friendly" format while paying attention to "customer" requirements provided to you.

Your program must compile using jGrasp. Any final program that does not compile, for any reason, will receive an automatic zero. Other IDEs often place in additional code that you are unaware of, doing too much of the work for you. **You are strongly discouraged from using IDEs other than jGrasp.**

Grading Criteria	
Requirement	**Points**
Implementation of object-oriented Java program, using efficient practices, such as the use of constants, good variable names, information hiding, no redundant code, etc.	70
Appropriate objective-style documentation	10
Appropriate intermediate comments	20