NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
ISLAMABAD CAMPUS

CS-1004 Object Oriented Programming Fall-2021

ASSIGNMENT-04

Sections (All)

Submission Deadline: 02 December, 2021 (11:59 pm)

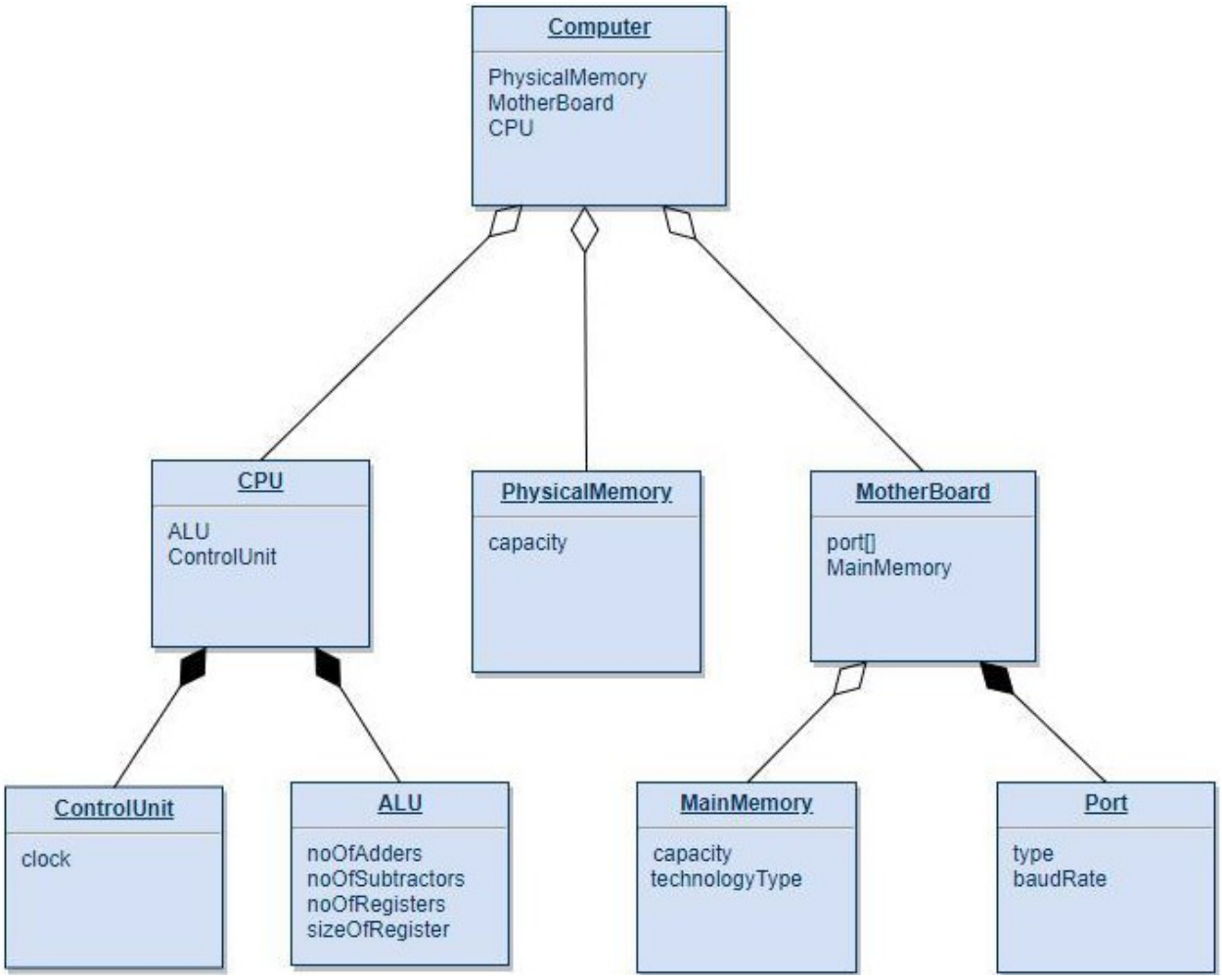
Instructions:

1. This is an individual assignment. You must complete this assignment by yourself. You cannot work with anyone else in the class or with someone outside of the class. The code you write must be your own and you must understand each part of your code. You are encouraged to get help from the instructional staff through Google Classroom.
2. **Test cases:** Test cases are not mandatory. However, test cases will have bonus marks other than 100 marks of the assignment. Test cases will be shared with you in Google Classroom.
3. **Driver:** Write a separate *main()* function as a driver for each question. The driver should demonstrate all the required functionalities of a question.
4. **Source Code:** Your code must be *generic*, *well-commented*, and *well-organized*. It should not have compilation errors. Each class should have private data members. Each class should also contain constructors (*default*, *parametrized*, and *copy constructor* if needed), a destructor, setters, and getters. Each class should be implemented in separate .h and .cpp files.
5. **Marks Distribution:** Total marks of the assignment are 100. There are 3 questions. Marks distribution for each question is mentioned.
6. **Submission Guidelines**: Submit assignment according to the guidelines given below.
   a) Combine all your work in one .zip file. Use a proper naming convention for your submission file. Name the .zip file as ROLL-NUM_SECTION (e.g., 20i-0001_A). Your zip file should contain separate folders for each question, but no subfolders within each folder. Each folder should contain the .h and .cpp files having necessary (#ifndef) header guards.
   b) Run and check your program on a lab machine before submission. If there is a syntax error, zero marks will be awarded in that specific question.
   c) Submit the .zip file on Google Classroom within the deadline.
   d) Submission other than Google classroom (e.g., email, etc.) will not be accepted. If we cannot download the file from Google classroom due to any reason it will lead to zero marks in the assignment.
7. **Queries:** Understanding is part of your assignment so avoid unnecessary queries. You can post your queries (if not asked by anyone before) on Google Classroom.
8. **Plagiarism**: Plagiarism of any kind (copying from others, copying from the internet, etc.) is not allowed. If found plagiarized, you will be awarded zero marks in the assignment. Repeating such an act can lead to strict disciplinary actions and failure in the course.

Note: Make sure that you have read and understood every instruction before starting the assignment.

Question 1: 30 marks

You need to write a host of classes, and place them in a reasonable hierarchy as shown below:



Description:

Design a class ALU which includes the following attributes:

- NoOfAdders: a int
- NoOfSubtractor: a int
- NoOfRegisters: a int
- sizeOfRegisters: a int

The class has the following member functions.

1. A default constructor.
2. A parameterized constructor.
3. Getters and Setters.

Design a class ControlUnit which includes the following:

- clock: a float

The class has the following member functions.

1. A default constructor.
2. A parameterized constructor.
3. Getters and Setters.

Design a class CPU which is composed of ALU and ControlUnit. Data members are:

- alu: an ALU
- cu: a ControlUnit

Part of relationship, Composition. (delete part when whole destroy).

The class has the following member functions.

1. A default constructor.
2. A parameterized constructor.
3. Getters and Setters.

Design a class MainMemory which includes the following:

- capacity: an int
- technologyType: a string (possible values: Semiconductor, Silicon)

The class has the following member functions.

1. A default constructor.
2. A parameterized constructor.
3. Getters and Setters.

Design a class Port which includes the following:

- type: a string (possible values: VGI Port, I/O Port, USB Port, HDMI Port etc.)
- baud_rate: an int

The class has the following member functions.

1. A default constructor.
2. A parameterized constructor.
3. Getters and Setters.

Design a class MotherBoard which is composed of Ports (IO ports, VGI ports etc) and aggregated with MainMemory:

- mm: A MainMemory
- ports: ports array

Part of relationship, Composition. (delete part when whole destroy).

The class has the following member functions.

1. A default constructor.
2. A parameterized constructor.
3. Getters and Setters.

Design a class PhysicalMemeory which includes the following:

- capacity: an int

The class has the following member functions.

1. A default constructor.
2. A parameterized constructor.
3. Getters and Setters.

Design a class Computer which is aggregation of PhysicalMemory, CPU and MotherBoard, and includes the following:

- pm: A PhysicalMemory
- mb: A MotherBoard
- cpu: A CPU

Aggregation, independent classes.

The class has the following member functions.

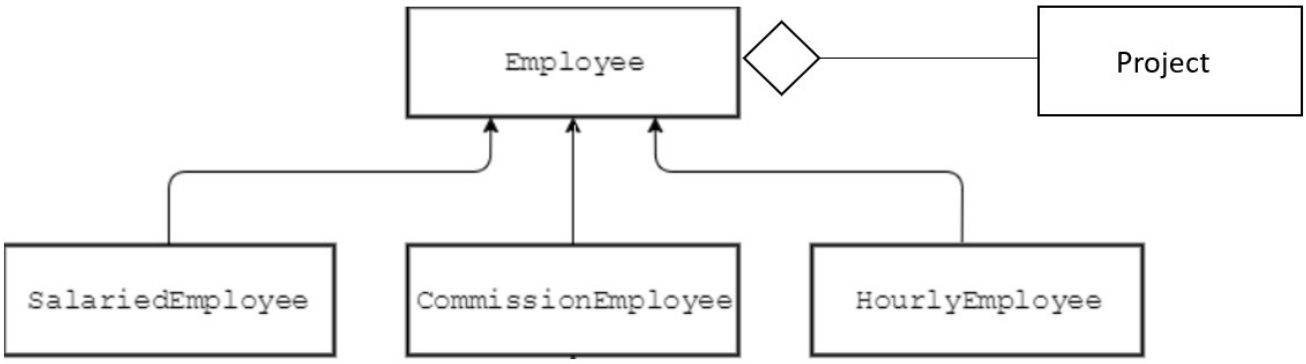
1. A default constructor.
2. A parameterized constructor.
3. Getters and Setters.

To Do:

To test these classes, create a Shop class which manufactures and sells computers. In the Shop class, write a function manufactureComputer() which creates a new Computer object by taking all the necessary specifications from user and adds it to a Dynamic Array of Computers. The user can then view list of the Computers which were manufactured.

Question 2: 30 marks

Consider the following UML Diagram.



In the above given diagram, it can be seen that "Salaried Employee", "Commission Employee" and "Hourly Employee" are inherited from the "Employee" class. (Recall the concepts of inheritance)

Assume that each employee is assigned a project so each employee has a project on which he is working on. (Recall the concepts of Aggregation)

The attributes and methods for each class are given in the table below.

Class	Attributes	Functions
Project	Project Id (Integer) Project Details (String)	Constructor Getters and Setter for Project Id and Project Details showDetails()
Employee	Name (String) Employee Id (String) Project (Project class' object)	Getters and setters for name and employee id showDetails()
Salaried Employee	Monthly Salary (Integer)	Getter and setter for salary

		getSalary() showDetails()
Commission Employee	Gross Sales (Integer) Base Salary (Integer) Total salary (Float)	Getter and setters for gross salary and base salary getTotalSalary() calculateSalary() showDetails()
Hourly Employee	No of hours (Integer) Hourly rate (Integer) Total Salary (Integer)	Getters and setters for number of hours and hourly rate. calculateSalary() getTotalSalary() showDetails()

Make sure to use the data encapsulation techniques, where necessary. The details for each class are given below.

private and public methods

Project:

Each project has a "project ID" and "project details" (write any random details about the project here).

There is **non-parameterized constructor** to instantiate a Project object. Using getters and setters, get the values and set the "id" and "project details". The ID must be at least 1 or greater than this.

"Show details" function must **return** the Project ID and Project Details **as a string.**

Employee:

There is a "name", "employee id" and a "project" associated with each employee.

To convert any data type to string

https://www.educative.io/edpresso/how-to-convert-an-int-to-a-string-in-cpp

validate: check for correct val
(id should be <= 1)

There are getters and setters that should be used to get and set the values of "employee name" and "id" and should validate the input too. The ID cannot be less than 1. "Show details" function must return the Employee ID, name, salary, Project ID and Project Details **as a string.**

Salaried Employee:

Getter and setter are used to set the "monthly salary". Monthly salary must be above Rs. 30,000.

"GetSalary" returns the monthly salary as an integer. "Show details" function must **return** the Employee ID, name, salary, Project ID and Project Details **as a string.**

Commission Employee:

Gross sales show the total sales achieved by the employee. Getters and setters are used to get and set the gross sales and base salary values.

Base salary must be above 20,000. Gross sales must be a positive integer value.

1.5% Commission is only given to the employee if the gross sales exceed Rs. 100, 000 and is calculated using the formula:

*Commission = gross sales * 0.015*

"calculateSalary" function calculates the total salary of the employee and sets the "total salary" attribute to the calculated salary using the following formula:

Total salary = base salary + commission

"GetTotalSalary" returns the total salary as a floating number. "Show details" function must **return** the Employee ID, name, Project ID and Project Details **as a string.**

Hourly Employee:

Getters and setters are used to retrieve and set the values of "no of hours" and "hourly rate". No of hours worked must be at least 30 and the hourly rate must be at least Rs. 150

"calculateSalary" function calculates the total salary using the given formula and sets the "total salary" attribute to the result of the calculation:

*Total salary = no of hours * hourly rate*

"GetTotalSalary" returns the total salary as an integer. "Show details" function must **return** the Employee ID, name, salary, Project ID and Project Details **as a string.**

To Do:

Write a driver code to test all the functions; you can hardcode the values instead of taking the inputs from the user.

You can also make other utility functions as needed if that makes your program simpler and/or easier to read. Your program should not crash on the edge cases or illegal inputs.

Question 3: 40 marks

In this Question, you will design and develop an automated management system for a Walk-through Restaurant where customers can place their orders and pay the bill to receive ordered items. Read the case study carefully and identify all the classes necessary to develop the system and put them in a reasonable hierarchy.

Order:

Info
Date
Time
TotalBill
Status
GTotal

The system enables the user to view the orders placed by customers. For each order, the system stores customer information, date and time at which the order was placed, total bill and status (pending, in-progress, ready). For each Order, 5% tax is included in the grand total.

Menu:

Id
Name
Price

Customers can put orders by selecting items from the list of available items. For this, the restaurant must use customer's name and phone number. Customer can place order and make payment in the system. The restaurant menu is organized in categories (main course, beverages, and desserts) of menu items. Each menu item has an id, name, and price in Rs.

The main course menu includes sandwiches, pizzas and a further sub-category of fried items. For each main course item, customer can provide a description (special instructions). Sandwich can be specified by its filling (smoked chicken, BBQ chicken, chicken jalapeno, vegetable) and type (Grilled/Plain).The restaurant offers pizza in four different sizes (small, medium, large, extra-large) with three different crust options (Fresh pan, Stuffed and Hand-tossed) and many topping options (Pepperoni, Mushrooms, Onions, Extra cheese and Black olives).

The Fried items sub-category includes nuggets, fries and burgers. Each fried item is available with many sauce options to choose from. The restaurant offers three different types of nuggets (Chicken Nuggets, Tempura Nuggets and Crispy Chicken Nuggets) in three different shapes (circle, star and heart) with dipping Sauces (Sweet and Sour, Honey Mustard, Sweet Chili and Hot Mustard). Two types of fries are available: curly and plain in three different servings (Small, Regular, and Jumbo) with different sauces (Garlic-Mayo, Chocolate and Cheddar Cheese). To order a Burger, a customer can select from different options available for Patties (Beef, Crispy Chicken, Grilled Chicken and fish), Bun Type (plain and sesame seed), and Burger Sauces (Jalapeno cheese sauce, Classic burger sauce, cheese mayo and Truffle mayo).

The restaurant offers beverages of different flavors in three different servings (Small, Regular, Large) and in two different temperature levels (i.e. Room temperature, Chill). Beverages category is further divided in two sub categories (Carbonated Drinks, Fresh Juices).

Carbonated Drinks (Sprite, 7up, Pepsi, Coke, Red Bull and Fanta) can be specified by name and whether or not the drink is sugar free. Fresh juices have two types (Fresh/Pre Packed) and can further be specified whether or not it's seasonal.

Dessert menu contains a variety of flavors of ice creams and cookies. The system stores Price per scoop for all the available ice creams. Ice cream price is calculated with numbers of scoops and customer can choose from two types to toppings (Wafers and Peanuts). Cookies are available in different flavors (Chocolate chip, Peanut, Coconut). The system stores price per gram for each cookie type. The price of the cookies is calculated by price per gram and is also available as Sugar free. Cookies are available in different shapes (Round and Square).

The system takes an order by displaying the menu to the customer. Each order can be composed of multiple items and the system generates and displays the total bill to the customer. Customer can then proceed with the order or cancel the order. If customer chooses to proceed with the order, the system displays the bill. Customer can then pick the order from the restaurant by paying the bill. System should be able to display all the cancelled orders and ready orders.

Payment is handled manually outside the system. System will only display the grand total to the customer. When a customer successfully completes an order, the system changes the status to ready immediately and displays a success message to the customer.

To Do:

Provide the UML class diagram for your design. Provide driver code to test the developed system.