CS 5800

1) INHERITANCE – "IS A" Relationship

Create the classes in the inheritance hierarchy given below. Each class should have appropriate constructors, set methods and get methods, also known as accessors and mutators respectively.

An **Employee class** should have a first name, last name and social security number.
A **SalariedEmployee class** should have a weekly salary.
A **HourlyEmployee class** should have a wage and a number of hours worked.
A **CommisionEmployee class** should have a commission rate and gross sales.
A **BaseEmployee class** should have a base salary.

Finally, write a driver program, that is a program that will instantiate each of the classes above and populate with the following information:

First name	Last name	Social Sec #	Weekly Salary	Wage	Hours worked	Com rate	Gross salary	Base salary
Joe	Jones	111-11-1111	$2,500					
Stephanie	Smith	222-22-2222		$25	32			
Mary	Quinn	333-33-3333		$19	47			
Nicole	Dior	444-44-4444				15%	$50,000	
Renwa	Chanel	555-55-5555	$1,700					
Mike	Davenport	666-66-6666						$95,000
Mahnaz	Vaziri	777-77-7777				22%	$40,000	

2) POLYMORPHISM

Design a **Ship class** that has the following members:

1. A member variable for the name of the ship(String)
2. A member variable for the year that the ship was built(String)
3. Constructor with all setters and getters(Also known as accessors and mutators)
4. A **print function** that will display the ship name and year it was built.

Design a **CruiseShip class** that is derived from the **Ship class** and has the following members:

1. A member variable for the maximum number of passengers(Int)
2. Constructor with all setters and getters(Also known as accessors and mutators)
3. A **print function** that overrides the base print function and will display the ship name and maximum number of passengers.

Design a **CargoShip class** that is derived from the **Ship class** and has the following members:

1. A member variable for the cargo capacity in tonnage(Int)
2. Constructor with all setters and getters(Also known as accessors and mutators)
3. A **print function** that overrides the base print function and will display the ship name and ships cargo capacity.

Demonstrate the classes in a program that has a static **array of Ship** size of 3.. The array elements should be initiated dynamically with one type each of **Ship**, **CruiseShip**, and **CargoShip objects**. Create a loop that iterates through the array calling each print function.

3) AGGREGATION - "HAS A" Relationship

You will create a Course class that "has a" Instructor and "has a" textbook. The Instructor and textbook should be there own type(That means each is a class). In the Course class you will instantiate the instructor and textbook objects as needed.

Create a **Course Class** which holds the following information:

1. Course *name*(private String)
2. An instructors first name, last name and office number. This information is accessed using the setter and getters of the instructor class.
3. The textbooks title, author and publisher. This information is accessed using the setter and getters of the texbooks class.
4. Print function that prints out Course name, instructor first and last name and the text book title and author.
5. Create a driver program to show this information. You can use our course information: Instructor Nima Davarpanah office 3-2636, textbook Clean Code…etc
6. Now modify the code to have two instructors and two textbooks.
7. Create a driver program to show this information, make up the other information.

4) COMPOSITION - “OWN A” Relationship

You will create a **Folder class** and a **File class** that will represent the File System on your computer. They should have all the proper member variables, setters and getters. The folder class will have a print function that prints out all subfolders and files. The file class will have a print function that prints out the name of the file.

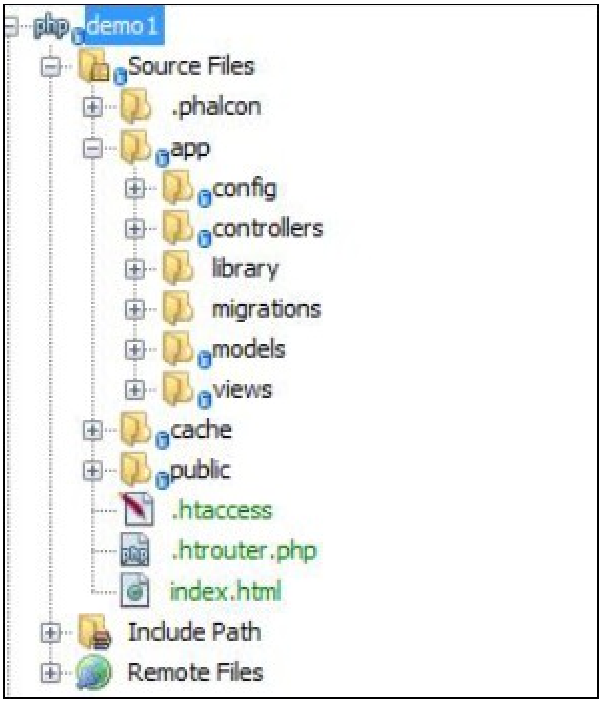
A Folder can contain zero or more Files. A Folder can also contain zero or more Sub-Folders.

A File must be stored within a folder.

If a folder is deleted all the files and sub-folders in the folder are also deleted.

In the structure below we see **php_demo1** is a folder and has a sub-folder of **Source Files.**

Recreate the structure below:



```
php_demo1
    Source Files
        .phalcon
        app
            config
            controllers
            library
            migrations
            models
            views
        cache
        public
        .htaccess
        .htrouter.php
        index.html
    Include Path
    Remote Files
```

1. As part of the driver program print out the php_demo1 folder and all sub folders. All sub folders should also print out their content until the full structure above is printed.
2. Now delete the folder **app** and print out the full structure.
3. Now delete the folder **public** and print out the full structure.