

SWE4005

OBJECT ORIENTED PROGRAMMING

Contents

Part A (Report).....	1
Part B (consists of three parts)	1
Portfolio 2 (Part A Description - Class Relationships).....	2
Portfolio 2 (Part B Description - Inheritance/Interfaces)	4
Portfolio 2 (Part C Description - Design Patterns)	4

Part A (Report) : In a brief report (1300-1500 words) summarize the basic principles of object-oriented program design and explain the basic ideas behind class hierarchies, polymorphism, and programming to interfaces. discuss the challenges of studying software engineering and development in a professional and business context (link this context to the application problem of part 2 project assessment)

Part B (consists of three parts)

Construct a solution to a given application problem by going through the basic steps of program specifications, analysis, design, implementation and testing --- within the context of the object-oriented paradigm.

Your solution should be presented in class. For this reason, you should create a brief presentation using MS Powerpoint or other program/ web platform, describing also the software requirements of the particular case.

The presentation should include the names of the creator, module name and topic

Application problem:

The goal of the project is to create a web or desktop application (i.e. e-shop, e-hospital or a e-service) where the student will apply several object-oriented concepts such as (Inheritance , Encapsulation, Polymorphism, Abstraction, Aggregation, Association, Composition, SOLID Principles). Furthermore, student will have to implement inside his project one or more design patterns like Singleton or Factory Pattern.

Portfolio 2 (Part A Description - Class Relationships)

You have the following classes [DOCTOR, PATIENT, ADDRESS, ROOM]

Doctor			Address		
	Id	int		Id	int
	FirstName	string		Name	string
	LastName	string		Country	string
	Age	int		City	string
	Salary	double		Postal Code	string
	List of Patients	Patient - list		List of Patients	Patient - list
Patient			Room		
	Id	int		Id	int
	FirstName	string		Title	string
	LastName	string		List of Patients	Patient - list
	Age	int			
	EntryDate	Date Time			
	ExitDate	Date Time			
	One Address	Address			
	One Room	Room			
	One Doctor	Doctor			

A Doctor can have many Patients

A Patient can have one Address, one Room, one Doctor

An Address can have many Patients

A Room can have many Patients

Make an application that does the following:

A1 Populates with enough Data (Synthetic Data - written in your code) the following Tables. Perhaps 10-15 data inside every table is enough data. Also relate every entity according to the above assumptions

1. Doctor
2. Address
3. Patient
4. Room

A2 Print with acceptable format the above (without their relationships & Ids)

Example of printing Doctors

First Name	Last Name	Age	Salary
George	Papas	35	65012
Peter	Sellers	55	59000

As you can see, I did not print patients or address

So, your application will print The Following Tables

1. Doctor
2. Address
3. Patient
4. Room

A3 Also Print the following related tables

1. All patients per Room
2. All patients per Doctor
3. All patients per Address
4. All addresses per patients

First Name	Patients
George Papas	
	Doe Fontan
	Ballou Bulldog
Peter Sellers	

	Edgar Poe
	Leonardo Winston

Example of printing patients per doctor

A4 Finally

Finally build a small application (like a menu) where the user can read the above information and also create a filter bar in each table where user will be able to search for data according to a property (e.g.: find all doctors with a salary greater than 10000 or by name)

Portfolio 2 (Part B Description - Inheritance/Interfaces in class Project)

By using Inheritance & Polymorphism in your project or in the sample-project named “karantina” given to you in class, your goal is to make a refactoring, so we can find all the employees by Salary.

To make the above work in your project, just add another entity Nurse with the same properties and relations as doctor.

Portfolio 2 (Part C Description - Design Patterns in class Project)

A) By using Singleton Design Pattern add a logging functionality in your project