C# Programming Homework 07

Chapter 07, C# Step by Step

Readings

Read chapter 7 in the C# Step by Step book.

Discussion Questions

Answer the discussion questions in writing.

1. What is a class? According to the book, what does a class arrange?"

Root word of classification, a type. Systematically arrange information and behavior into a meaningful entity.

1. What are the two purposes of encapsulation?

To allow a program to create an instance of a class and calls the methods of that class.

1. How do you instantiate an instance of a class? How do you access that instance?

Call up or declare the variable representing the class

1. What is the default access of the fields and methods of a class? How do you change the default?

The default access is private, can be used by other methods inside the class only. With the public and private keywords.

1. What is the syntax for writing a constructor?

It has the same name as the class, and it can take parameters, but it cannot a value (not even void)

1. What is the difference between class fields and methods, and instance fields and methods? How do you

create class fields and methods?

Class fields and methods you don’t have to call, instance fields and methods you do.

1. How do you bring a static class in scope? Why would you want to bring a static class in scope?

You can call the method or access to the field by using the name of the class.

1. Can you think of a good reason to create an anonymous class? What is it?

When you already have the name of an instance

9. What is polymorphism as this term is used in computer science? This is not in the book.

10. What is message passing as this term is used in computer science? This is not in the book.

11. What was the first object-oriented programming language?

SImula

12. Consider this quote by Alexander Stepanov:

I \_nd OOP technically unsound. It attempts to decompose the world in terms of interfaces

that vary on a single type. To deal with the real problems you need multisorted algebras

| families of interfaces that span multiple types. I \_nd OOP philosophically unsound. It

claims that everything is an object. Even if it is true it is not very interesting | saying that

everything is an object is saying nothing at all.

Who is Alexander Stephanov? What do you think about this quote?