Bilgisayar Mühendisliği Bölümü

YDM313 Mesleki İngilizce II

1. Arasınav, 12.11.2009

Group A

ANSWERS

1. **OVERLOADING** provides more than one method with the same name but different 'method signatures' as a convenience to calling method. **(5P)**

2. **Method**s can also be marked with the visibility modifier. When methods are **PUBLIC** they are callable from the code that also reference the instance of that class. When methods are marked as **PRIVATE** they are usually considered what are called 'Hepler Methods' or rather methods that perform some small part of overall function of the public methods in your class. They can be use to break down the function of one big method into smaller byte sized methods. **(10P)**

3. **INHERITANCE** is a tennet of OOP that allows a class to derive its attributes and its functionality from a base class. **(5P)**

4. Did you ever vonder why we never had to define an instance of **MessageBox** class before using the **Show** method of the **MessageBox** object all over the place. The reason is because the **Show** method has been defined with the **STATIC** keyword which means that is not necessary to define an instance of the class before using that particular method. The same is true whenever we use the **Int.Parse** statement. **(10P)**

5. Let's focus on the use of modifiers in terms of the classes' properties, fields and methods. When we design a class we will want the certain things to be visible and certain things to be hidden to the code that uses an instance of that class. Those things that are marked with the keyword **PRIVATE** will be the things that are not accesible by the code that defined instance of our object. Now in the case of **field** like we defined in our car class, the only way that the code can retrieve and use the value that stored in the these **fields** is by using its associated **PROPERTY** and notice that those are marked as public. **(10P)**

6. I defined public class `Minivan : Car`. This little notation of the colon ":" means that there's relationship between two classes. Car is a **BASE** class. **(10P)**

7. I changed the visibility to something called **PROTECTED** . This allows those classes that derived from the base class to access **private** fields. **(10P)**

8. We have used the same name for our method that we used for our class. `class Car{...}` and our method name is public `Car(){...}`. The purpose of the **CONSTRUCTOR** is that we can use that to initialize values. **(5P)**

9. The blueprint is not a house. The recipe is not a cookie. They define how the each of should be performed or built but it is not the same thing. You can build several houses from the same buleprint and each of the houses is an instance of the blueprint. A **CLASS** is the blueprint and **OBJECT** is an instance of that blueprint. **(10P)**

10. The derived class can specialize the base class by **OVERRIDING** methods. **(10P)**

11. The property has two distinct pieces: You have a **GET** statement that allows to return the field back when its requested and you write code to in the **SET** statement to assign the underlying filed value to whatever is being assigned to. **(10P)**

12. Scroll down in your game window until you find `protected override void` **DRAW** . This method is another XNA Framework method. It is called repeatedly every time the game needs to draw a frame. **(5P)**

DO NOT FORGET TO WRITE YOUR NAME AND NUMBER