**In-Class Quiz [Classes, Objects, Data Abstraction and Encapsulation] 3/27/2020**  
**Student Name: Allen Adams**

1. **(3pts) What is OOP? What are the benefits of OOP? When should OOP be used?**

Object oriented programing is the process of using classes manipulate data through the creation of object instances. The benefits of oop can be observed in reusability of code as well as in efficiency of data structure. Oop should be used whenever possible to cut down on project overhead. More specifically oop is best used in projects that demand the manipulation and handling of data.

1. **(3pts) What is a Class? What are the benefits of creating a class?**

In oop a class is an encapsulation of code that can exist outside of the main program executable. One purpose can be to separate and better organize code. However, it also allows for reusability and cuts down on redundant programing.

1. **(3pts) What is an Object?**

An object represents an instance of a class. An object has perimeters that define what it is made of. An example of an object could be a car. A car has a make, model and color… in OOP a car can be represented by an object with the same parameters that define make, model and color.

1. **(3pts) What is a Constructor? What is it used for?**

A constructor is like a recipe for an object. The constructor of an object defines what parameters need to be satisfied for the object to be properly represented and created.

1. **(3pts) What are class Properties? When do we use them?**

A property of a class is a variable that can be accessed throughout the class itself. These properties can be get or set and depending on the type of encapsulation can be accessed throughout the entire program.

1. **(3pts) What are class methods? When do we use them?**

Methods allow for the manipulation of class objects. Methods allow you to access or change information pertaining to an object once its been created. Methods are used when we want to read or write any data pertaining to an object.

1. **(3pts) What is data Abstraction? Why is it important?**

Data abstraction involves the process of capturing similar information among data entities and making objects based on those commonalities. This is important because it can reduce the overhead of the projects exponentially. An example of this might be you want to make an application that tracks different types of cards (as in playing cards, trading cards ect). Instead of making several class objects for each type of card, you can make one class covers all bases and captures the similarities among all cards such as name, size, color (these sorts of things).

1. **(3pts) What is Encapsulation? Why is it important?**

Encapsulation is the process of governing the accessibility of classes, and class variables. You can determine weather certain classes should be public or private and who/ what should access them through encapsulation. This is important because it allows the programmer to protect certain information and control restrictions on data.

1. **(3pts) What are accessors? When do we use them? Describe an example for each type of accessor and when to use it.**

An accessor is like a lock set on classes and class variables. These can be used to deny the editing or viewing of certain information from outside references. For instance, the public accessor, allows other classes to reference certain information from the class that has this accessor attached to it. Here you could reference a data value from another class to perhaps display the data or change it elsewhere. The private accessor makes it so the class can only be referenced by itself. Here, you can only change data values within the same class, and it will not be accessible via outside classes. This can be used for security reasons or when handling sensitive information.

1. **Create three classes:**
   1. **(20pts) PassFailCourse:** This class should describe a course that has a name and a pass/fail grade. This class should have a method to return the grade. Make sure the grade cannot be changed after it has been set for the first time.

**Class PassFailCourse{**

**Public string Name;**

**private bool PF;**

**public PassFailCourse(string name, bool pf){**

**Name = name;**

**PF = pf;**

**}**

**public bool getGrade(){**

**return PF;**

**}**

**}**

* 1. **(25pts) GradedCourse:** This class should describe a course that has a name and a decimal grade value. This class should have a method to return the grade as passed if >=60, otherwise it is a failed course. Make sure the grade cannot be changed after it has been set for the first time.

**Class GradedCourse{**

**Public string Name;**

**private double Grade;**

**public PassFailCourse(string name, double grade){**

**Name = name;**

**Grade = grade;**

**}**

**public bool getGrade(){**

**if (Grade >=60){ return true; }**

**}**

**Else {return false;}**

**}**

**}**

* 1. **(30pts) Degree:** This class should describe a degree that has four courses: two are grades and two are pass/fail courses. This class should have a method that says a degree is fulfilled (passed) if at least three out of the four courses are passed. There should be another method that provides GPA (set a pass course = A and failed course = F). You also need to create a method that returns how many classes have been passed, and another method that counts how many classes the student failed.

**Class Degree{**

**Public PassFailCourse PFA, PFB;**

**Public GradedCourse GCA, GCB;**

**public PassFailCourse(GradedCourse gca, GradedCourse gcb, PassFailCourse pfa, PassFailCourse pfb){**

**GCA = gca;**

**GCB= gcb;**

**PFA = pfa;**

**PFB = pfb;**

**}**

**public int numPassed(){**

**int numPass =0;**

**If(PFA.getGrade() == true){ numPass++;}**

**If(PFB.getGrade() == true){ numPass++;}**

**If(GCA.getGrade() == true){ numPass++;}**

**If(GCB.getGrade() == true){ numPass++;}**

**Return numPass;**

**}**

**Public bool Passed(){**

**If (numPassed() >= 3){return true;}**

**Else{return false} }**

**Public double getGPA(){**

**Return (double)(numPassed());**

**}**

**Public int numFailed(){**

**Return (numPassed() – 4);**

**}**

**}**

* 1. **(13pts)** Write some code that creates a degree object with appropriate courses then display the degree outcome and GPA.

**PassFailCourse MastersCourse1 = new PassFailCourse("Masters Course 1", true);**

**PassFailCourse MastersCourse2 = new PassFailCourse("Masters Course 2", false);**

**GradedCourse MastersCourse3 = new GradedCourse("Masters Course 3", 86);**

**GradedCourse MastersCourse4 = new GradedCourse("Masters Course 4", 63);**

**Degree MastersDegree = new Degree(MastersCourse1, MastersCourse2,**

**MastersCourse3, MastersCourse4);**

**Console.WriteLine(MastersDegree.Passed());**

**Console.WriteLine(MastersDegree.getGPA());**

**Console.ReadKey();**