**Name: Patrick O’Brien**

For each of the below questions, write a short sentence or two to express (in your own words) your answer. Keep the answers short, but use complete, correct, English sentences.

If it helps to clarify the questions, feel free to mentally prefix all the questions with the phrase "According to the video…"

1. After you’ve watched all the videos, please answer this question:  
   Of all the videos that you watched, if you could pick one video to be re-recorded by the instructor outside of class which would you choose? Why?  
   (Keep in mind the recording outside of class will omit any pauses from the instructor answering student questions, have less hemming and hawing, etc, and generally be more concise)

|  |
| --- |
| The first video OOP\_Basics does not cover anything about a rectangle class as described in the quiz questions. I would make sure that this video and quiz covers the correct content. |

**VIDEO: OOP Basics**

1. The basic idea behind object-oriented programming is to combine what two things together, in one place?

|  |
| --- |
| Yes |

1. What is the first “part” of the rectangle class?

|  |
| --- |
| Car c = new Car(); 🡪 creating a new object to be called |

1. Data should always be declared as what?  
   (Remember that you can always create a public method to access these data fields later.)

|  |
| --- |
| Data should be declared as actions |

1. Even though the behavior (the methods) are defined for all instances of this class, calling the calculateArea method will cause the area to be calculated based on what?

|  |
| --- |
| OOP\_basics video does not have a method that calculates area. It is based on data and action and cars. |

1. What is the “constructor” used for?

|  |
| --- |
| A constructor is a special method of a class or structure in object-oriented programming that initializes an object of that type.  The constructor usually has the same name as the class. It can be used to set the values of the members of an object, either to default to user-defined values. - Techopedia |

1. How does the constructor copy a starting value for the width (or height) into the rectangle’s “permanent” memory?

|  |
| --- |
| This is not covered in the video. I assume it’s because that it creates a temporary object, and then adds the temporary object to the “permanent” memory. |

1. Given a choice between making a variable a permanent (“instance”) variable or a temporary variable (typically a local variable), which type should you prefer to use? Under what circumstances would you use the other one?

|  |
| --- |
| I would use a temporary/local variable in a situation where it would only be used in the local method. If it isn’t used anywhere else, there’s no need to allocate it to permanent memory. I would therefore use permanent/instance variables in situations where it may be used between multiple methods or classes. |

**VIDEO: OOP Encapsulation**

1. In C# you can use either a getHeight() method or a C# property. Why will we be using the getHeight() method in this course?

|  |
| --- |
| We are using getHeight() because we can use it with java, c++, c#, etc. It’s universal |

1. In addition to allowing us to make basic changes to objects (such as changing **r1**’s height to be 10), what else does the accessor method make it easy for us to do?  
   (Hint: What if we try to change **r1**’s height to be **-**10)

|  |
| --- |
| It makes it easy to change the value that we are using to calculate the height in r1. Makes it easy to change the object referred to by r1. If we feed it bad number like -10, it won’t accept it, we can filter it. |

1. If one makes the height field public what implications will that have for the program (in terms of who can access that field)?

|  |
| --- |
| Anyone anywhere can at any time modify the height variable. Any code from any coworkers could potentially modify the data. It is not a security measure, it just lets people modify the values from anywhere in the program. |

1. Is the public/private/protected feature that we’re looking at a security mechanism?

|  |
| --- |
| It is not a security mechanism. |

1. Why are the constructor and calculateArea methods allowed to access the private **height** field?

|  |
| --- |
| Because the constructor is a method within the overall class. CalculateArea is also within the class. |

1. Can the **main**  method access the private **height** field of the **r1** object?

|  |
| --- |
| No it cannot. |

1. Why is it ok to call the calculateArea method outside the Rectangle class (say, in **main**)?

|  |
| --- |
| Because it is set to public, it can be accessed outside the class or anywhere in the code. |

1. Let’s say that you wanted to ensure that the height field of all rectangle objects are not negative. Prior to the public/private/protected access-control mechanism, what was your only option for ensuring that this would be true? Why would this be difficult?

|  |
| --- |
| The only way would be to do an if else check. You might throw and exception, logging the error. It is difficult because you would have to think up every potential bug-causing input. Basically it is impossible without encapsulation. |

1. Fill in the blank: “Just to recap, encapsulation is:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”

|  |
| --- |
| Marking data fields as public or private to restrict access to specific components of software. Generally everything should marked private unless you have a really good reason for it not to be. |

**VIDEO: Object Oriented Programming (OOP): Background / Motivation**

1. Prior to object oriented programming, how were programs (typically) divided up?

|  |
| --- |
| This video is not listed on the website.  Prior to object oriented programming, programs were divided up by:  Procedural programming paradigm. An example is languages like c++ that are lower level and closer to machine code. The design method used in procedural programming is called top down design.  Top Down Designs is when you start with the problem and break it down into sub-problems. This is functional decomposition. This continues until the problem is straightforward enough to be solved by a sub-procedure.  Software maintenance using this method can be difficult and time consuming. When changes are made to the top procedure, changes can cascade down to sub procedures of main, and their sub-procedures. Making a small change somewhere in the program can lead to cascading failure. |

1. How does object oriented programming (typically) change the way programs are divided up?

|  |
| --- |
| Object oriented programming instead takes main modules and separates them into classes rather than procedures. This separates critical components of the program into separate components that can be called individually. It is then easier to modify and debug complex programs. |

1. Fill in the blanks: “The basic idea is that in object oriented programming we’ll have a [Car] **class, which describes \_\_\_\_\_\_\_\_\_\_\_\_\_\_**, and we’ll have an individual **object which describes \_\_\_\_\_\_\_\_\_\_\_\_\_\_**”

|  |
| --- |
| The video isn’t listed in the website so I can’t answer specific questions like this. |

1. Fill in the blank: “An object is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_” ( about 6:30)

|  |
| --- |
| An object refers to an instance of a class. An object can be a combination of variables, functions, and data structures. |

**VIDEO: OOP: (Instance) Methods, Object Allocation Demo**

1. What is a good type of name for a class (and what are a couple of examples of bad names for a class)?

|  |
| --- |
| A good name typically describes what the class does. Separating things into different namespaces and packages. |

1. If you wanted to add a “Print” command to a class named Car, where would you put the Print method’s definition? Describe the location in English, then copy the example code from the video (at about the 1:50 mark)

|  |
| --- |
| You would put the print method definition in the class Car, to be called from the main class.  Class Car  {  Public void Print()  {  Console.WriteLine(“Car object”);  }  }  Class Program  {  Static void Main(string[] args)  {  Car c = new Car();  c.Print();  }  } |

(After 2:15 the video answers a number of tangential questions –skim this if you’re interested, but don’t worry about recalling the details of this part of the video)

**VIDEO: OOP: Instance Variables**

1. Fill in the blank: “Unless you’ve got a really good reason [to do otherwise], you should always make all your instance variables \_\_\_\_\_\_\_\_\_\_\_\_\_”

|  |
| --- |
| private |

1. When you create a new Car object, what value will be put into all of the instance variables for that particular object?

|  |
| --- |
| All the instance variables will be given a default value for their type. |

1. Describe in your own words how the SetMaxSpeed method allows one to change the maximum speed for a car object.

|  |
| --- |
| The SetMaxSpeed takes a value in the form of a parameter. Take the value of newMax and copy it into the value of the one particular car. |

1. Why is it ok to set maxSpeed in the SetMaxSpeed method, and then use it in the Print method?

|  |
| --- |
| Because it is an instance variable that is defined outside the method. maxSpeed is a private int, which retains its value as the program updates it. |

1. How does writing out as your variable as **this.**maxSpeed help to make your program more clear?

|  |
| --- |
| This.maxSpeed  C= 70,000  When object created, default values are filled in. newMax=100 in SetMaxSpeed()  Invisible parameter “this”  Is going to be whatever object you called it on in class Program.  This.maxSpeed = newMax was 70,000, but is now changed to 100 since newMax=100 |

(Starting at about 9:30 there’s a very visual explanation of how memory is managed when creating and using objects. Even though there aren’t a ton of questions about this, definitely watch this in order to develop an intuition about how object creation & usage works!!)

1. What is the purpose of the invisible parameter named **this**? What information does it pass from main to the SetMaxSpeed method?

|  |
| --- |
| The purpose is to add new methods without having to recompile the original type.  Public void SetMaxSpeed(int maxSpeed)  {  (This.maxSpeed 🡪 the instance variable) = (maxSpeed 🡪 the parameter)  } |

1. Inside the SetMaxSpeed method, how does C# figure out what the (variable) name maxSpeed refers to?

|  |
| --- |
| In the case of two maxSpeed variables, one refers to the parameter, while the other refers to the instance variable. |

1. Because of the procedure you outlined in your answer to the prior question, is it possible to create a local variable that has the same name as an instance variable?

|  |
| --- |
| Yes it would be possible to do so, however you shouldn’t, as this would lead to confusion and possible errors in the program. |

1. Should you ever create a local variable that has the same name as an instance variable?   
   ( Hint: NO!!!!!!!!!!!!!! ☺ )

|  |
| --- |
| No |

1. Does the method’s name cause the action/effect of the method?   
   If not, why is it still a good idea to choose method names that describe the each method does?

|  |
| --- |
| No it does not. The method’s name is only used for reference for the programmers. It is a good idea to use it so can quickly know what each method is doing. |

1. The “Stack” is very, very efficient at doing what?

|  |
| --- |
| The stack is very efficient at allocating and deallocating. |

**VIDEO: OOP: Access control (public/private), getter/setter methods**

1. The primary purpose of access control is prevent my co-workers from doing what?

|  |
| --- |
| Accidently modifying the important values in your code. For instance if you share the same public variable name between different classes, they could perform an operation that modifies the variable that you are depending on to a certain value. By making it private and using encapsulation this cannot happen. |

1. What can the SetMaxSpeed method do that will help prevent errors for being made in the overall program?

|  |
| --- |
| You can force coworkers to use the SetMaxSpeed method to change/check the value of maxSpeed. This way you can more accurately control what the expected input is. |

1. What does the keyword “private” mean / what does it do?

|  |
| --- |
| Private is a part of access control/encapsulation. It restricts the access to the private component so that it can only be accessed from inside the class that it was created. |

1. If you try to access a private data member (say, in **main**), what error message will you get?

|  |
| --- |
| Main is inaccessible due to its protections level. |

1. Why is **main** allowed to call SetMaxSpeed? Why is SetMaxSpeed allowed to change **maxSpeed**?

|  |
| --- |
| Because SetMaxSpeed is a public method that can be accessed from any class. It is allowed to change the value of maxSpeed because maxSpeed is declared within the same class, and SetMaxSpeed is a method within that class. |

1. Another advantage to forcing everyone to use accessor methods is that you can change what part of the class/program and yet NOT have to change the rest of the program?

|  |
| --- |
| A property accessor is used to return the property value. A set property accessor is used to assign a new value. The part that you can change is the accessor accessibility level. |

(Around the 9 minute mark there’s a blurb about exception handling – it is interesting to know about, but you’re not required to know it (unless there’s other material in this class that specifically requires you to know it) ).

1. Does access control protect you from intentionally malevolent actions by your co-workers? If not, then where is it useful?

|  |
| --- |
| No, access control would not protect you from hostile internal intent in any way. It is useful for unintentional edits of your code. For instance, if public instance variable names, or method names are shared between parts of the program. Having private components keeps classes contained, so data cannot be accidently changed from other classes. |

**VIDEO: Overloading**

1. Are over**load**ing a method and over**riding** a method the same thing? If not, give a really quick, intuitive explanation of each one (you haven’t seen overriding yet so don’t worry about providing a detailed/entirely correct explanation – the main thing is that you know these are different)

|  |
| --- |
|  |

1. There’s an example of overloading in the video involving a printBox method (right around the 3 minute mark) – briefly summarize what we were trying to accomplish with the printBox methods, and why it’s useful to have multiple methods that all have the same name.

|  |
| --- |
|  |

1. The basic rule for figuring out if you’ve legally overloaded a method is if you’ve got enough information at what point in the program? What information do you need, specifically?

|  |
| --- |
|  |

1. Fill in the blank: “Overloading makes sense when you have one \_\_\_\_\_\_\_\_\_\_\_, but a whole bunch of different \_\_\_\_\_\_\_\_\_\_\_\_”

|  |
| --- |
|  |

1. Can C# differentiate which version of the method to call based on the return value?  
   If not, why not?

|  |
| --- |
|  |

1. Is it less efficient to call one overloaded version of the function, and then have it immediately call the ‘real’ version? If so, why do we do it?

|  |
| --- |
|  |

1. Give an example of an overloaded method that’s built into the .Net Framework Class Library:

|  |
| --- |
|  |

**VIDEO: OOP: Constructors**

1. Implementing a constructor for your class will help you avoid making what sort of error?

|  |
| --- |
|  |

1. When does the constructor method run/execute?

|  |
| --- |
|  |

1. When the Car class has only 1 constructor, that takes a single parameter, what sort of compile-time can occur? Why is this error good / useful?

|  |
| --- |
|  |

1. What two things identify a method as being a constructor?

|  |
| --- |
|  |

1. How can you have one constructor call a different constructor? Give a brief snippet of C# code that demonstrates this, and then provide a brief explanation (in English, in your own words) about how this works:

|  |
| --- |
|  |

**VIDEO: Basic Array of Objects**(You can jump to the 1:30 mark & watch from there – the first minute is not useful ☺ )

1. In your own words BRIEFLY outline what the **new** operator does when asked to create an array of Demo objects. Make sure that you’re clear what value (if any) that gets placed into each array slot.

|  |
| --- |
|  |

1. What does the keyword **null** mean/represent in C#?

|  |
| --- |
|  |

1. What is a simple, intuitive rule for remembering what default value will be filled into a new array?

|  |
| --- |
|  |

1. In the below snippet of C# code, what does line **(1)** do? What does line **(3)** do? Do you really need both? How are they different?

Demo [] ds = new Demo[ 5 ]; **(1)**  
Console.WriteLine( ds[ 0 ] );  
ds[1] = new Demo(); **(3)**

|  |
| --- |
|  |

1. How do you call the printBox() method on the object at array slot #1 in the array named ds?

|  |
| --- |
|  |

**VIDEO: Null References Within An Array**

1. What happens when you attempt to call the printBox() method on the third slot in the ds array (ds[2]) ? Why does that happen?

|  |
| --- |
|  |

1. What sort of exception will see in when you attempt to call the printBox() method on the third slot in the ds array (ds[2]) ?

|  |
| --- |
|  |

1. How can you check (using C# code) if an array slot is **not** null? Provide a short snippet of C# code, and a brief, intuitive explanation of how it works:

|  |
| --- |
|  |

1. How can you check (using C# code) if an array slot **is** null? Provide a short snippet of C# code, and a brief, intuitive explanation of how it works:

|  |
| --- |
|  |