**OOP Concepts Assignment - Solutions**

1. [3] Describe the term OBJECT and give a “real world” example.   
   An item or thing that will have **specific** states and behaviors and models a “real world” object.  
   examples: like the bicycle analogy (a particular mountain bike that is blue and has 10 speeds, etc), or a house (a specific townhouse that has an address and people living in it), or a phone (a specific model of a phone, a Samsung Galaxy S9 model #12345), a dog object would be a specific dog named Rover, whose breed is Labrador, color is brown, weight is 90 pounds, etc.
2. [2] Describe the term CLASS and give a “real world” example.  
   Template for an object that will define what states and behaviors any object can have  
   examples: similar to a **blueprint** for a house, or a **pattern** for a garment, etc..
3. [4] How does Object Oriented programming differ from Procedural programming? \*\*There’s not JUST one difference!   
   Procedural → defines data then has actions, instructions or subprograms that act on that data, top-down instructions, complex code, difficult to edit/fix  
   OOP → objects created that encapsulate data then methods act on the objects, modular (code reuse) and building block style, complex design, less difficult (easier) to edit/fix
4. [2] Based on the following statement, state the OBJECT, its states (attributes) and behavior (methods).  
   *“Compute the perimeter of a rectangle given its length and width.”*Object → rectangle  
   Attributes → length, width   
   Method → perimeter (computing)  
   \*\*\*Note: It’s not wrong to say perimeter is also an attribute, but if you have a method to calculate it, you don’t really NEED it as an attribute.
5. [2] How many new objects can be created in one class? Explain.  
   As many as a programmer needs (technically an infinite amount), space (heap size) being the only possible issue to restrict the amount of objects
6. [2] What are the 2 **essential** elements that **must** be in ANY OOP program in order for it to execute?  
   The **main** method and a public **class** that contains this main method
7. [3] What are the 3 important items that make up a **class heading** (NOT the body / declarations!) in Java code?   
   <modifiers> such as public, private, void or a data type  
   The key word CLASS  
   <identifier> - a name given by the coder in the form of:  
   **<modifiers> class <identifier> {}**
8. [3] What is the difference between a CLASS method (or attribute) and an INSTANCE method (or attribute)? What is the keyword that defines a class method (or attribute)?  
   A class method or attribute belongs to the whole program and all objects. The attributes can be used anywhere and methods can be called without an object. The keyword static defines class attributes and methods. Instance attributes and methods belong to an object and can only be used with a particular object. Unique “instances” (specifications/values) of that attribute are set when an object is created and in order to manipulate instance attributes, instance methods are needed.
9. [4] Name and briefly describe some different **categories** (NOT examples) of methods. (Try to find 3 or 4, other than the ***class*** and ***instance*** methods mentioned in question 10)  
   Constructor methods: used to assign values to object attributes ONLY  
   Helper methods: used to help code efficiency, such as something that could repeat often throughout code  
   Function (value-returning method): can be a static method that will return a value or a factory method that returns an instance of the native class  
   Accessor methods: basically getter methods that simply return the value of an instance (usually private) attribute  
   There are possibly a few differently named ones depending on your resources found. (Factory methods, non-returning, void or procedural methods, setters/getters (accessor) - in general)
10. [4] Name and briefly describe some (at least 3 or 4) **modifiers** (could be used with classes, methods or variables) in Java. (You probably have to look this up.)  
    Access modifiers: public, private, protected - determine how or from where the code can be “seen” and/or used  
    Non-access modifiers: void, static, final, abstract, transient, volitile - determine other things about the identifier, such as non-returnable, immutable, or even for variables and functional (value returning) methods a data type is basically a modifier (int, float, String, char, etc)
11. [4] What is the API (for Java)? Basically, what does it stand for? Where can it be found (web address)? And, what is the general purpose of it?  
    The **A**pplication **P**rogramming **I**nterface for Java is at:  
    <https://docs.oracle.com/javase/8/docs/api/index.html>  
    It outlines all the packages, classes, attributes, methods, interfaces, exceptions, etc. that are defined in the Java programming language and used by any programmer. It is a software program interface that aids a programmer in building efficient programs.  
    Most languages have some form of API which outlines all a languages’ specifications, protocols, commands, etc.
12. [7] Define the following OOP terms briefly (in one point or sentence):
    1. Attributes  
       Attributes are properties and are like the adjectives or descriptors of a class/object
    2. Behaviours (Methods or Responsibilities)  
       Behaviours are methods and are like the verbs or actions of the class/object
    3. Encapsulation  
       Data hiding by the bundling of attributes and methods into a single unit. Data (variables/attributes) are hidden and can only be accessed through methods of the current class. Its PURPOSE is to provide security.
    4. Instantiation  
       The act of creating an object, based on its class. A constructor is called to assign values (default or otherwise) to the object’s attributes.
    5. Abstraction  
       Abstraction is a process of hiding the implementation details from the user, only the functionality will be provided to the programmer. In other words, the programmer will have the information on what the object does instead of how it does it. From: www.tutorialspoint.com (The definition of abstract is dealing with an idea not any specifics; theoretical over practical)
    6. Interface  
       An interface is a collection of empty abstract methods implemented by a class to make sure the methods are described in the class. It is like a **contract** with the “outside” world. It is like an **agreement** among a team of programmers that shows the names/identifiers and basic structure of classes, attributes and methods that are used throughout the program.
    7. Package  
       Similar in concept to a folder on a computer system, this is a container for all the files necessary for a Java application to run. (Such as class files, images, text files, etc.)