1. Compare and contrast elements of procedural PHP vs OOP PHP. Do you prefer procedural or OOP? Why?

*Procedural PHP focuses primarily on functions that manipulate data, while OOP focuses on structuring data into objects that are easier to conceptualize. I myself prefer OOP not only because it is what I have worked with the most, but also because it is easier to keep track of how the data in the software should behave.*

1. List and Define 4 OOP Concepts.

*Inheritance: Objects can be created as sub-classes of other objects and inherit the attributes of the superclass. For example, a superclass of Dog may have a bark() function that is inherited by all sub-classes, thus allowing the Yorkie class access to bark().*

*Encapsulation: Restricting access to portions of the program’s logic strictly to the classes by which they are implemented by.*

*Abstraction: Abstraction is very similar to encapsulation where complex logic is hidden from the parts of the program that don’t need it, but abstraction takes it a step further and instead grants a class access to a method that references logic that is contained safely away in another part of the code.*

*Namespacing: A data structure that prevents the clashing of names or data from different parts of the code, because variables and data that would normally collide, such as id numbers, are recognized to belong to the namespace in which they are instantiated, and are thus distinct.*

1. How would you instantiate a new object in PHP?

*A new Object is instantiated with the use of the new keyword, and the name of the class to be created.*

1. Write a line of PHP code that takes an object and references its properties. You can make up the names of the object and the properties.

*$myProject = Project::user->id(5);*

1. What is a constructor? What does it do? Show an example in PHP code.

*A constructor defines the attributes of a class when it is instantiated.*

*class User {*

*public function \_\_construct(protected int $id, protected string $username) {*

*$id = $this-id;*

*$username = $this->username; }*

*}*

1. What is a destructor? What does it do? Show an example in PHP code.

*A destructor is a function that is called when the are no remaining instances of a class.*

*function \_\_destruct() {*

*print\_log(‘There are no more users online, shutting down’);*

*exit();*

*}*

1. What is a class? What is a method? Show an example of each in PHP code.

*A class is an object definition that contains the methods and relationships that the object has access to.*

*Class Post {*

*protected $fillable = ['body', ‘image’];*

*Public function user()*

*{ Return $this->belongsTo(User::class); }*

*}*

*$postedBy = Post::user();*

1. What is encapsulation?

*Encapsulation is the idea of a class encapsulating all of the logic it needs to perform it’s job internally without having to expose that logic to other parts of the software that don’t require access to that logic, just the functionality. Much like a driver does not need to understand an internal combustion engine to drive a car.*

1. Show an example of PHP syntax for creating a MySQLi object.

*$mysqli = new MySQLi(DB\_HOST, DB\_USER, DB\_PASSWORD, DB\_NAME);*

1. What is an exception?

*An exception is the OOP equivalent of a procedural code error. When an object encounters an issue, it throws an exception, which can be caught in a try/catch block to execute user (or developer) friendly messages or additional logic.*