**Question 1**

**1 / 1 pts**

There are four principles of Object Oriented Programming. Which principle allows a class to get and use all the public fields and methods of another class?



Abstraction



Encapsulation

**Correct!**



Inheritance



Polymorphism

**Question 2**

**0 / 1 pts**

Which principle of Object Oriented programming says that an object should be able to change to reflect different uses and environments?

**You Answered**



Abstraction



Encapsulation



Inheritance

**Correct Answer**



Polymorphism

**Question 3**

**0 / 1 pts**

Which principle of object oriented programming says that an object should be as general as possible, and that ideally a field should not repeat but be located in a single place?

**Correct Answer**



Abstraction

**You Answered**



Encapsulation



Inheritance



Ploymorphism

**Question 4**

**1 / 1 pts**

Which principle of Object Oriented Programming says that a class should be as self contained and complete as possible?



Abstraction

**Correct!**



Encapsulation



Inheritance



Polymorphism

**Question 5**

**1 / 1 pts**

In class relationships an Association

**Correct!**



Occurs when one class talks to another by calling a method in the other class



occurs when a class is a child of another class and will receive use of all the parents public methods and fields



occurs when a class contains instances of the another class, but the contained class has a separate existence



occurs when a class is entirely contained in another object and disappears when that object is destroyed

**Question 6**

**0 / 1 pts**

Composition

**You Answered**



Occurs when one class talks to another by calling a method in the other class



occurs when a class is a child of another class and will receive use of all the parents public methods and fields



occurs when a class contains instances of the another class, but the contained class has a separate existence

**Correct Answer**



occurs when a class is entirely contained in another object and disappears when that object is destroyed

**Question 7**

**0 / 1 pts**

Inheritance

**You Answered**



Occurs when one class talks to another by calling a method in the other class

**Correct Answer**



occurs when a class is a child of another class and will receive use of all the parents public methods and fields



occurs when a class contains instances of the another class, but the contained class has a separate existence



occurs when a class is entirely contained in another object and disappears when that object is destroyed

**Question 8**

**0 / 1 pts**

Aggregation

**You Answered**



Occurs when one class talks to another by calling a method in the other class



occurs when a class is a child of another class and will receive use of all the parents public methods and fields

**Correct Answer**



occurs when a class contains instances of the another class, but the contained class has a separate existence



occurs when a class is entirely contained in another object and disappears when that object is destroyed

**Question 9**

**0 / 1 pts**

Which of the following are true of inheritance?



A class that inherits gets all the public methods and fields of the parent class

**You Answered**



Inheritance moves from the more general to the more specific: Person to Customer



You can only inherit from a class if the child is of the same type: Customer from Person, not Sale from Person

**Correct Answer**



All of the above

**Question 10**

**1 / 1 pts**

This is an example of



Association

**Correct!**



Inheritance



Aggregation



Composition

**Question 11**

**0 / 1 pts**

This represents an example of



Association



Inheritance

**You Answered**



Aggregation

**Correct Answer**



Composition

**Question 12**

**0 / 1 pts**

This represents an example of

**Correct Answer**



Association



Inheritance



Aggregation

**You Answered**



Composition

**Question 13**

**0 / 1 pts**

This represents an example of

**You Answered**



Association



Inheritance



Aggregation

**Correct Answer**



Composition

**Question 14**

**1 / 1 pts**

class Employee(Person):

This python code shows an example of which relationship below?



the class has an association with Person

**Correct!**



The class Employee is inheriting the class Person



person is variable being passed from another class



None of the above

**Question 15**

**1 / 1 pts**

In python the \_\_init\_\_() method

**Correct!**



Is the class constructor that sets the initial conditions of the class



returns the a string when the class is cast to string



is just another method



None of the above

**Question 16**

**1 / 1 pts**

An python the \_\_str\_\_() method



initializes the class values



casts a value to string

**Correct!**



returns a string when the class is cast to string such as in a print metnod



All of the above

**Question 17**

**0 / 1 pts**

Which best defines a class field?



A school baseball field

**Correct Answer**



A class level variable that defines some aspect of the class

**You Answered**



An action the class can perform



None of the above

**Question 18**

**0 / 1 pts**

In Python a class field is identified



by where it is declared

**Correct Answer**



by using the "self" keyword to show it belongs to the class

**You Answered**



by including it in a function



None of the Above

**Question 19**

**1 / 1 pts**

public enum Suits {

A getter or accessor method is one



That allows a class or user to change a variable's value

**Correct!**



allows a class or user to retreive a class value



hides a class value



None of the above

**Question 20**

**0 / 1 pts**

This type of diagram is called a

**You Answered**



Use Case Diagram



Activity Diagram



Class Diagram

**Correct Answer**



Sequence Diagram

**Question 21**

**0 / 1 pts**

The square boxes represent



Time lines

**Correct Answer**



Objects

**You Answered**



Actors



Messages

**Question 22**

**0 / 1 pts**

The arrows represent



Objects

**You Answered**



Time Lines

**Correct Answer**



Messages



Actors

**Question 23**

**0 / 1 pts**

The dashed vertical lines represent

**Correct Answer**



Time Lines

**You Answered**



Messages



Nothing



Everything

**Question 24**

**1 / 1 pts**

def test\_lockString(self):

lock=Lock(3176,'be','normal')

self.assertEqual(str(lock),'3176 locked')

This code represents an example of (use best answer)



An lock of some kind



A call to a function

**Correct!**



A unit test



None of the above

**Question 25**

**1 / 1 pts**

When doing unit testing you should



Make sure all calculations are correct



Check what happens when the wrong values are entered



Check for all exceptions

**Correct!**



All of the above