**Task Title: Task Title: Week 3 Python Task: OOP, Classes and Objects**

* **Elective**
* **Full Stack 1: Frontend And Python**

**Created 04-10-2022 17:47:01**

**Read the following questions and provide the correct answers:**

1. \_\_\_\_\_ represents an entity in the real world with its identity and behaviour.

A.) A class

B.) A method

C.) An object

D.) A reference

2. \_\_\_\_\_ is used to create an object.

A.) A class

B.) A function

C.) A constructor

D.) A method

3. Class “Football” inherits from the class “Sport”. How do I correctly specify this in my python code?

A.) from Sport import Football

B.) Football = Sport()

C.) class Football(Sport)

D.) class Sport(Football)

4. Which of the following does not correctly create an object instance?

A.) ferrari = Car(“white”)

B.) benz = Car(“red”)

C.) bmw = new Car(“yellow”)

D.) honda = Car()

5. What does the following code output?

class People():

    def \_\_init\_\_(self, name):

      self.name = name

    def print\_name(self):

      print(self.name)

  person1 = People("Gregory")

  person2 = People("Mike")

  person1.print\_name()

A.) person1

B.) Gregory

C.) Mike

D.) Gregory Mike

6. What will be the output of the following Python code?

class test:

     def \_\_init\_\_(self,a=""Welcome to Zuri""):

         self.a=a

     def display(self):

         print(self.a)

obj=test()

obj.display()

A.) The program has an error because constructor can’t have default arguments.

B.) Nothing is displayed on the console

C.) The program has an error display function doesn’t have parameters

D.) “Welcome to Zuri” is displayed

7. What is Instantiation in terms of OOP terminology?

A.) Copying an instance of a class

B.) Creating a class

C.) Creating an instance of a class

D.) Modifying an instance of a class

8. The \_\_\_\_\_\_\_\_\_ keyword defines a template indicating the data that will be in an object of the class and the functions that can be called on an object of the class.

A.) init

B.) class

C.) instance

D.) Class

9. \_\_\_\_\_\_\_\_\_ is not a keyword, but by convention, it is used to refer to the current instance (object) of a class.

A.) init

B.) self

C.) def

D.) class

10. What is the output of the following code?  
class Pokemon():

    def \_\_init\_\_(self, name, type):

        self.name = name

        self.type = type

    def stringPokemon(self):

        print(f"Pokemon name is {self.name} and type is {self.type}")

class GrassType(Pokemon):

    # overrides the stringPokemon() function on 'Pokemon' class

    def stringPokemon(self):

        print(f"Grass type pokemon name is {self.name}")

poke1 = GrassType('Bulbasaur', 'Grass')

poke1.stringPokemon

poke1.stringPokemon()

poke2 = Pokemon('Charizard', 'Fire')

poke2.stringPokemon

poke2.stringPokemon()

  A.) Error because the extending class has a stringPokemon() function which already exists.

 B.) Grass type pokemon name is Bulbasaur Grass type pokemon name is Charizard

 C.)  Pokemon name is Bulbasaur and type is Grass Pokemon name is Charizard and type is Fire.

D.) Grass type pokemon name is Bulbasaur Pokemon name is Charizard

 and type is Fire

ANSWER

C

C

C

C

B

D

C

B

B

D