1. Which of the following represents a distinctly identifiable entity in the real world?

A. A class
**B. An object**
C. A method
D. A data field

2. What will be the output of the following code snippet?

```
class Sales:
    def __init__(self, id):
        self.id = id
        id = 100

val = Sales(123)
print (val.id)
```

A. SyntaxError, this program will not run
B. 100
**C. 123**
D. None of the above

Reason : we are just trying to access the id variable which relates to the sales class. To access it we use self.id.
Here id = 100 means we are actually overwriting the id parameter, it won't affect self.id

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

A. puppy = Dog("Jamie")
B. dog = Dog("Jamie")
C. jamie = Dog()
**D. pupper = new Dog("Jamie")**

Reason : no keyword named "new" is available in python to create an object instance.

4. What does the following code output?

```
class People():
    def __init__(self, name):
      self.name = name

    def namePrint(self):
      print(self.name)
```

```
person1 = People("Emma")
person2 = People("Watson")
person1.namePrint()
```

**A. Emma**
B. Watson
C. Emma Watson
D. person1

Reason : we are just printing the person1's name, that is Emma

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

A. class
B. def
**C. self**
D. init

6.Which of the following is the correct way to define an initializer method?

A. def __init__(title, author):
**B. def __init__(self, title, author):**
C. def __init__():
D. __init__(self, title, author):

Reason : to initialize a constructor in python, we must use self in the parameter.

7.Which of the following represents a template, blueprint, or contract that defines objects of the same type?

**A. A class**
B. An object
C. A method
D. A data field

Reason : A class in a template or blueprint that creates an object

8.
```
class Dog:
    def __init__(self, name, age):
        self.name = name
        self.age = age
```

The correct way to instantiate the above Dog class is:

**A. Dog("Rufus", 3)**

B. Dog()
C. Dog.__init__("Rufus", 3)
D. Dog.create("Rufus", 3)

Reason : our constructor takes two parameters.

9.In Python, a function within a class definition is called a:

**A. a method**
B. a class function
C. a callable
D. an operation

10.

```
class Person:
    def __init__(self, id):
        self.id = id

sam = Person(100)
sam.__dict__['age'] = 49

print (sam.age + len(sam.__dict__))
```

A. 1
B. 2
C. 49
**D. 51**

Reason : here sam.age = 49 and length of sam.dict is 2