ENGR 101 Introduction to Programming

Study Questions - Week 12

Q1- Write an init method for the Vehicle class that takes x, y and z as optional parameters and assigns them to the corresponding attributes(colour, speed, brand).

Q2-

```
class Pokemon:
    health = 0
    attack = 0


pikachu = Pokemon()
pikachu.health = 100
pikachu.attack = 15

print "Pokemons attack is : " +str(pikachu.attack)+ "   Pokemons health is : "
+str(pikachu.health)
```

Read the above code and tell the output.

Q3- Create 4 Pokemon objects, which has health, attack, name attiributes. Do not use init function while creating them.

Q4-

```
class Pokemon:
    def __init__(self, name, health, attack):
        self.PokemonsName = name
        self.PokemonsHealth = health
        self.PokemonsAttack = attack

pikachu = Pokemon("Pikachu", 100, 15)
balbasaur = Pokemon("Balbasaur", 115, 10)
charmander = Pokemon("Charmander", 90, 20)
squirtle = Pokemon("Squirtle", 105, 12)
```

Read the code and tell what it does.

Q5- Create 10 pokemons with name, health, attack and type attiributes. Type of the pokemon could be electricity, fire, water and plant. Use __init__ function while creating them.

Q6- Give your 2 pokemon objects to the function given below as inputs. What is the output?

```
def fight_arena(poke1, poke2):
    round = 1
    while poke1.PokemonsHealth > 0 or poke2.PokemonsHealth>0:
        poke1.PokemonsHealth = poke1.PokemonsHealth-poke2.PokemonsAttack
        poke2.PokemonsHealth = poke2.PokemonsHealth-poke1.PokemonsAttack
        print "Round :"+str(round)+"  " +poke1.PokemonsName +" vs " +
poke2.PokemonsName
        print poke1.PokemonsName+ "  hit "+ str(poke1.PokemonsAttack) + " damage
" + \
              poke2.PokemonsName +"'s " + str(poke2.PokemonsHealth) + " health
left"
        print poke2.PokemonsName + "  hit " + str(poke2.PokemonsAttack) + "
damage " + \
              poke1.PokemonsName + "'s " + str(poke1.PokemonsHealth) + " health
```

```
left"
        round += 1
        if poke1.PokemonsHealth < 0:
            print poke2.PokemonsName + " WINS !!!"
            break
        elif poke2.PokemonsHealth <0:
            print poke1.PokemonsName + " WINS !!!"
            break
fight_arena(pikachu, charmander)
```

Q7- Modify the fight_arena function so each Pokemon attacks would be multiplied by a random float number between 0 and 1.

Q8- Create an animal class then create 3 objects of animal class with some attiributes.

Q9- In the lecture you wrote a class that represents a point in 2D space. Now, outside of the class, write a function, move, that will take a point object p, integers dx and dy, and move p by the dx and dy.

```
class Point:
    '''represents a point in 2D'''
    def __init__(self,x,y):
        self.x = x
        self.y = y
```

Q10- Define a class Student that will have as attributes name, surname, department, semester and gpa. Create an instance of this class and print the attributes.

Q11- Define a class Date, that will have as attributes day, month, year (as integers). Give as default values 1 for each attribute. Now, outside of the class, write a function, print_date, that will take a Date object, and print the date in this format: day/month/year.

Q12- Define a class Triangle that will have as attributes its three internal angles (angle1, angle2, angle3). Now, outside of the class, write a function, triangle_check, that will check if a given Triangle object/instance is a triangle or not. That is, it should return True if the sum of three angles on the Triangle object is equal to 180 and False, otherwise.

Q13- Define a class Vehicle that will have as attributes the type of vehicle, the number of wheels, engine size, and its color. Color's default value should be red. Now, outside of the class, write a

function that will change the color of a given Vehicle instance to a given color value. Create an instance with the default color value, then change it using your function to blue.

Q14- Define a class Employee that will have as attributes name and salary. Whenever you create a new employee instance it should print out "You added a new employee, their name is: <given name>".

Q15- Define a class Car that will have as attributes speed- which is the current speed of the Car, and max_speed- which is the Car's maximum speed. Now, outside of the class, write a function, increase_speed that will increase the current speed of a given Car object for the given amount as an argument without exceeding the max_speed. If it exceeds the max_speed then set the current speed to equal to the max_speed.

Q16- Consider the class Car in question 15. Write a new function decrease_speed which will take a Car object, and another argument indicating the value for which to decrease the current speed. If the current_speed after the decrease is smaller then 0 set the current speed equal to 0.

Q17- Define a class named Hotel that can be initialized with total number of rooms and number of rooms occupied, respectively. Now, outside of the class, write a function, occupancy that takes a Hotel object, and returns the current occupancy percentage (number of rooms occupied/total number of rooms * 100).

```
>> myhotel = Hotel(150, 90)
>> print occupany(myhotel)
   The current occupancy is 60%.
```

Q18- Define a class RationalNum that represents rational numbers (numerator/denominator e.g. 2/3). Add an init method that initializes a rational number, given values for numerator and denominator. Your code should handle cases like 2/0 or 0/0 (both are not rational) and print an appropriate error message. Now, outside of the class, write a function, pretty_print that takes a RationalNum object, and prints the numerator and denominator as x/y.

```
>> rational = RationalNum(11, 23)
>> pretty_print(rational)
   11/23
```

Q19- Consider the RationalNum class in Q18. Now, write another function reduce that takes a RationalNum object and reduces the rational number to lowest terms. For example, 25/625 can be reduced to 1/25. See example below.

```
>> rational1 = RationalNum(60, 45)
>> print reduce(rational1)
   4/3
```