

October 3, 2022

```
[1]: #task_1

class Marks:
    def __init__(self, mark):
        self.mark = mark

    def __add__(self, obj):
        return Marks(self.mark+obj.mark)

Q1 = Marks(int(input("Quiz 1 (out of 10): ")))
Q2 = Marks(int(input("Quiz 2 (out of 10): ")))
Lab = Marks(int(input("Lab (out of 30): ")))
Mid = Marks(int(input("Mid (out of 20): ")))
Final = Marks(int(input("Final (out of 30): ")))
total = Q1 + Q2 + Lab + Mid + Final
print("Total marks: {}".format(total.mark))
```

```
Quiz 1 (out of 10): 10
Quiz 2 (out of 10): 10
Lab (out of 30): 30
Mid (out of 20): 20
Final (out of 30): 30

Total marks: 100
```

```
[2]: #task_2

class Teacher:
    def __init__(self, name, dept):
        self.__name = name
        self.__dept = dept
        self.__courses = []

    def addCourse(self, obj):
        self.__courses.append(obj.course_name)
```

```

def printDetail(self):
    print("=====")
    print("Name:", self.__name)
    print("Department:", self.__dept)
    print("List of courses")
    print("=====")
    for course in self.__courses:
        print(course)
    print("=====")

class Course:
    def __init__(self, course_name):
        self.course_name = course_name

t1 = Teacher("Saad Abdullah", "CSE")
t2 = Teacher("Mumit Khan", "CSE")
t3 = Teacher("Sadiah Kazi", "CSE")
c1 = Course("CSE 110 Programming Language I")
c2 = Course("CSE 111 Programming Language-II")
c3 = Course("CSE 220 Data Structures")
c4 = Course("CSE 221 Algorithms")
c5 = Course("CSE 230 Discrete Mathematics")
c6 = Course("CSE 310 Object Oriented Programming")
c7 = Course("CSE 320 Data Communications")
c8 = Course("CSE 340 Computer Architecture")
t1.addCourse(c1)
t1.addCourse(c2)
t2.addCourse(c3)
t2.addCourse(c4)
t2.addCourse(c5)
t3.addCourse(c6)
t3.addCourse(c7)
t3.addCourse(c8)
t1.printDetail()
t2.printDetail()
t3.printDetail()

```

=====

Name: Saad Abdullah  
Department: CSE  
List of courses

=====

CSE 110 Programming Language I

```

CSE 111 Programming Language-II
=====
=====
Name: Mumit Khan
Department: CSE
List of courses
=====
CSE 220 Data Structures
CSE 221 Algorithms
CCSE 230 Discrete Mathematics
=====
=====
Name: Sadia Kazi
Department: CSE
List of courses
=====
CSE 310 Object Oriented Programming
CSE 320 Data Communications
CSE 340 Computer Architecture
=====

```

```

[3]: #task_3

class Team:
    def __init__(self, name=None):
        self.__name = name
        self.__players = []

    def setName(self, name):
        self.__name = name

    def addPlayer(self, obj):
        self.__players.append(obj.player_name)

    def printDetail(self):
        print("=====")
        print("Team:", self.__name)
        print("List of Players:")
        print(self.__players)
        print("=====")

class Player:
    def __init__(self, name):
        self.player_name = name

```

```

b = Team()
b.setName('Bangladesh')
mashrafi = Player("Mashrafi")
b.addPlayer(mashrafi)
tamim = Player("Tamim")
b.addPlayer(tamim)
b.printDetail()
a = Team("Australia")
ponting = Player("Ponting")
a.addPlayer(ponting)
lee = Player("Lee")
a.addPlayer(lee)
a.printDetail()

```

```

=====
Team: Bangladesh
List of Players:
['Mashrafi', 'Tamim']
=====
=====
Team: Australia
List of Players:
['Ponting', 'Lee']
=====

```

[4]: #task\_4

```

class Color:
    def __init__(self, clr):
        self.clr = clr

    def __add__(self, obj):
        if((self.clr=="blue" and obj.clr=="red" ) or (self.clr=="red" and obj.
↪clr=="blue")):
            return Color("Violet")
        elif((self.clr=="yellow" and obj.clr=="blue" ) or (self.clr=="blue" and
↪obj.clr=="yellow")):
            return Color("Green")
        elif((self.clr=="red" and obj.clr=="yellow" ) or (self.clr=="yellow"
↪and obj.clr=="red")):
            return Color("Orange")
        else:
            pass

```

```

C1 = Color(input("First Color: ").lower())
C2 = Color(input("Second Color: ").lower())
C3 = C1 + C2
print("Color formed:", C3.clr)

```

First Color: red  
 Second Color: yellow  
 Color formed: Orange

```

[5]: #task_5

import math

class Circle:
    def __init__(self, radius):
        self.__radius = radius

    def getRadius(self):
        return self.__radius

    def setRadius(self, radius):
        self.__radius = radius

    def area(self):
        return math.pi*self.__radius*self.__radius

    def __add__(self, obj):
        return Circle(self.__radius+obj.__radius)

c1 = Circle(4)
print("First circle radius:" , c1.getRadius())
print("First circle area:" ,c1.area())
c2 = Circle(5)
print("Second circle radius:" ,c2.getRadius())
print("Second circle area:" ,c2.area())
c3 = c1 + c2
print("Third circle radius:" ,c3.getRadius())
print("Third circle area:" , c3.area())

```

First circle radius: 4  
 First circle area: 50.26548245743669  
 Second circle radius: 5  
 Second circle area: 78.53981633974483

Third circle radius: 9  
Third circle area: 254.46900494077323

```
[6]: #task_6

class Triangle:
    def __init__(self, base, height):
        self.__base = base
        self.__height = height

    def getBase(self):
        return self.__base

    def getHeight(self):
        return self.__height

    def setBase(self, base):
        self.__base = base

    def setHeight(self, height):
        self.__height = height

    def area(self):
        return 0.5*self.__height*self.__base

    def __sub__(self, obj):
        return Triangle(self.__base-obj.__base, self.__height-obj.__height)

t1 = Triangle(10, 5)
print("First Triangle Base:" , t1.getBase())
print("First Triangle Height:" , t1.getHeight())
print("First Triangle area:" ,t1.area())
t2 = Triangle(5, 3)
print("Second Triangle Base:" , t2.getBase())
print("Second Triangle Height:" , t2.getHeight())
print("Second Triangle area:" ,t2.area())
t3 = t1 - t2
print("Third Triangle Base:" , t3.getBase())
print("Third Triangle Height:" , t3.getHeight())
print("Third Triangle area:" , t3.area())
```

First Triangle Base: 10  
First Triangle Height: 5  
First Triangle area: 25.0

Second Triangle Base: 5  
Second Triangle Height: 3  
Second Triangle area: 7.5  
Third Triangle Base: 5  
Third Triangle Height: 2  
Third Triangle area: 5.0

[7]: #task\_7

```
class Dolls:
    def __init__(self, name, price, two_or_more = False):
        self.name = name
        self.price = price
        self.two_or_more = two_or_more

    def detail(self):
        if(self.two_or_more==False):
            return f"Doll: {self.name}\nTotal Price: {self.price} taka"
        else:
            return f"Dolls: {self.name}\nTotal Price: {self.price} taka"

    def __gt__(self, obj):
        if(self.price > obj.price):
            return True
        else:
            return False

    def __add__(self, obj):
        return Dolls(self.name+" "+obj.name, self.price+obj.price, True)

obj_1 = Dolls("Tweety", 2500)
print(obj_1.detail())
if obj_1 > obj_1:
    print("Congratulations! You get the Tweety as a gift!")
else:
    print("Thank you!")

print("=====")
obj_2 = Dolls("Daffy Duck", 1800)
print(obj_2.detail())
if obj_2 > obj_1:
    print("Congratulations! You get the Tweety as a gift!")
else:
    print("Thank you!")

print("=====")
```

```

obj_3 = Dolls("Bugs Bunny", 3000)
print(obj_3.detail())
if obj_3 > obj_1:
    print("Congratulations! You get the Tweety as a gift!")
else:
    print("Thank you!")

print("=====")
obj_4 = Dolls("Porky Pig", 1500)
print(obj_4.detail())
if obj_4 > obj_1:
    print("Congratulations! You get the Tweety as a gift!")
else:
    print("Thank you!")

print("=====")
obj_5 = obj_2 + obj_3
print(obj_5.detail())
if obj_5 > obj_1:
    print("Congratulations! You get the Tweety as a gift!")
else:
    print("Thank you!")

```

```

Doll: Tweety
Total Price: 2500 taka
Thank you!
=====
Doll: Daffy Duck
Total Price: 1800 taka
Thank you!
=====
Doll: Bugs Bunny
Total Price: 3000 taka
Congratulations! You get the Tweety as a gift!
=====
Doll: Porky Pig
Total Price: 1500 taka
Thank you!
=====
Dolls: Daffy Duck Bugs Bunny
Total Price: 4800 taka
Congratulations! You get the Tweety as a gift!

```

[9]: *#task\_8*

```

class Coordinates:
    def __init__(self, x, y):

```



```

        self.x = x
        self.y = y

    def __sub__(self, obj):
        return Coordinates(self.x-obj.x, self.y-obj.y)

    def __mul__(self, obj):
        return Coordinates(self.x*obj.x, self.y*obj.y)

    def __eq__(self, obj):
        if(self.x == obj.x and self.y == obj.y):
            return "The calculated coordinates are the same."
        else:
            return "The calculated coordinates are NOT the same."

    def detail(self):
        return f"({self.x},{self.y})"

p1 = Coordinates(int(input()),int(input()))
p2 = Coordinates(int(input()),int(input()))

p4 = p1 - p2
print(p4.detail())

p5 = p1 * p2
print(p5.detail())

point_check = (p4 == p5)
print(point_check)

```

1  
2  
3  
4

(-2,-2)

(3,8)

The calculated coordinates are NOT the same.

[ ]: