Question 1

In [1]:
```
class Student:
    def __init__(self, name='Just a student', dept='nothing'):
        self.__name = name
        self.__department = dept
    def set_department(self, dept):
        self.__department = dept
    def get_name(self):
        return self.__name

    def set_name(self,name):
        self.__name = name
    def __str__(self):
        return 'Name: '+self.__name+' Department: '+self.__department

class BBA_Student(Student):
    def __init__(self,name='default', dept='BBA'):
        super().__init__(name,dept)

print(BBA_Student())
print(BBA_Student('Humpty Dumpty'))
print(BBA_Student('Little Bo Peep'))
```

```
Name: default Department: BBA
Name: Humpty Dumpty Department: BBA
Name: Little Bo Peep Department: BBA
```

Question 2

In [1]:
```
class Vehicle:
    def __init__(self):
        self.x = 0
        self.y = 0
    def moveUp(self):
        self.y+=1
    def moveDown(self):
        self.y-=1
    def moveRight(self):
        self.x+=1
    def moveLeft(self):
        self.x-=1
    def __str__(self):
        return '('+str(self.x)+' , '+str(self.y)+')'

class Vehicle2010(Vehicle):

    def __init__(self):
        super().__init__()

    def moveUpperRight(self):
        self.x+=1
        self.y+=1

    def moveUpperLeft(self):
        self.x-=1
        self.y+=1

    def moveLowerRight(self):
        self.x+=1
        self.y-=1

    def moveLowerLeft(self):
        self.x-=1
        self.y-=1

    def equals(self,other):
        if (self.x == other.x) and (self.y == other.y):
            return "True"
        else:
            return 'False'
```

```
print('Part 1')
print('------')
car = Vehicle()
print(car)
car.moveUp()
print(car)
car.moveLeft()
print(car)
car.moveDown()
print(car)
car.moveRight()
print(car)
print('------')
print('Part 2')
print('------')
car1 = Vehicle2010()
print(car1)
car1.moveLowerLeft()
print(car1)
car2 = Vehicle2010()
car2.moveLeft()
print(car1.equals(car2))
car2.moveDown()
print(car1.equals(car2))
```

```
Part 1
------
(0 , 0)
(0 , 1)
(-1 , 1)
(-1 , 0)
(0 , 0)
------
Part 2
------
(0 , 0)
(-1 , -1)
False
True
```

Question 3

In [2]:
```
class Tournament:
    def __init__(self,name='Default'):
        self.__name = name
    def set_name(self,name):
        self.__name = name
    def get_name(self):
        return self.__name

class Cricket_Tournament(Tournament):
    def __init__(self,name = 'Default',team = 0,Type = 'No type'):
        super().__init__(name)
        self.__name = name
        self.team = team
        self.Type = Type

    def detail(self):
        return f"Cricket Tournament Name: {self.__name}\nNumber of Teams: {self.team}\nType: {self.Type}"

class Tennis_Tournament(Tournament):
    def __init__(self,name,player):
        super().__init__(name)
        self.__name = name
        self.player = player

    def detail(self):
        return f"Cricket Tournament Name: {self.__name}\nNumber of Players: {self.player}"

ct1 = Cricket_Tournament()
print(ct1.detail())
print("----------------------")
ct2 = Cricket_Tournament("IPL",10,"t20")
print(ct2.detail())
print("----------------------")
tt = Tennis_Tournament("Roland Garros",128)
print(tt.detail())
```

```
Cricket Tournament Name: Default
Number of Teams: 0
Type: No type
----------------------
Cricket Tournament Name: IPL
Number of Teams: 10
Type: t20
```

```
----------------------
Cricket Tournament Name: Roland Garros
Number of Players: 128
```

Question 4

In [3]:
```
class Product:
    def __init__(self,id, title, price):
        self.__id = id
        self. __title = title
        self. __price = price
    def get_id_title_price(self):
        return "ID: "+str(self.__id)+" Title:"+self.__title+"Price: "+str(self.__price)

class Book(Product):
    def __init__(self,id, title, price,num,publisher):
        super().__init__(id, title, price)
        self.__id = id
        self. __title = title
        self. __price = price
        self.num = num
        self.publisher = publisher

    def printDetail(self):
        return f"ID: {self.__id} Title: {self.__title} Price: {self.__price}\nISBN: {self.num} Publisher: {self.

class CD(Product):
    def __init__(self,id, title, price,band,time,genre):
        super().__init__(id, title, price)
        self.__id = id
        self. __title = title
        self. __price = price
        self.band = band
        self.time = time
        self.genre = genre

    def printDetail(self):
        return f"ID: {self.__id} Title: {self.__title} Price: {self.__price}\nBand: {self.band} Duration: {self.


book = Book(1,"The Alchemist",500,"97806","HarperCollins")
print(book.printDetail())
print("-----------------------")
cd = CD(2,"Shotto",300,"Warfaze",50,"Hard Rock")
print(cd.printDetail())
```

```
ID: 1 Title: The Alchemist Price: 500
```

```
ISBN: 97806 Publisher: HarperCollins
-----------------------
ID: 2 Title: Shotto Price: 300
Band: Warfaze Duration: 50minutes
Genre: Hard Rock
```

Question 5

In [4]:
```
class Animal:
    def __init__(self,sound):
        self.__sound = sound
    def makeSound(self):
        return self.__sound

class Printer:
    def printSound(self, a):
        print(a.makeSound())

class Dog(Animal):
    def __init__(self,sound):
        super().__init__(sound)

class Cat(Animal):
    def __init__(self,sound):
        super().__init__(sound)

d1 = Dog('bark')
c1 = Cat('meow')
a1 = Animal('Animal does not make sound')
pr = Printer()
pr.printSound(a1)
pr.printSound(c1)
pr.printSound(d1)
```

```
Animal does not make sound
meow
bark
```

Question 6

In [5]:
```
class Shape:
    def __init__(self, name='Default', height=0, base=0):
        self.area = 0
        self.name = name
        self.height = height
        self.base = base
    def get_height_base(self):
        return "Height: "+str(self.height)+",Base: "+str(self.base)

class triangle(Shape):
    def __init__(self,name='Default', height=0, base=0):
        super().__init__(name,height,base)

    def calcArea(self):
        self.area =  .5*self.height*self.base

    def printDetail(self):
        return f"Shape name: {self.name}\nHeight: {self.height}, Base: {self.base}\nArea: {self.area}"

class trapezoid(Shape):
    def __init__(self,name, height, base,side):
        super().__init__(name,height,base)
        self.side = side

    def calcArea(self):
        self.area =  ((self.side+self.base)/2)*self.height

    def printDetail(self):
        return f"Shape name: {self.name}\nHeight: {self.height}, Base: {self.base}, Side_A: {self.side}\nArea: {



tri_default = triangle()
tri_default.calcArea()
print(tri_default.printDetail())
print('--------------------------')
tri = triangle('Triangle', 10, 5)
tri.calcArea()
print(tri.printDetail())
print('--------------------------')
trap = trapezoid('Trapezoid', 10, 6, 4)
trap.calcArea()
```

```
print(trap.printDetail())
```

```
Shape name: Default
Height: 0, Base: 0
Area: 0.0
--------------------------
Shape name: Triangle
Height: 10, Base: 5
Area: 25.0
--------------------------
Shape name: Trapezoid
Height: 10, Base: 6, Side_A: 4
Area: 50.0
```

Question 7

In [6]:
```
class Football:
    def __init__(self, team_name, name, role):
        self.__team = team_name
        self.__name = name
        self.role = role
        self.earning_per_match = 0
    def get_name_team(self):
        return 'Name: '+self.__name+', Team Name: ' +self.__team

class Player(Football):
    def __init__(self,team_name, name, role,goal,match):
        super().__init__(team_name, name, role)
        self.__team = team_name
        self.__name = name
        self.goal = goal
        self.match = match

    def calculate_ratio(self):
        self.ratio = self.goal/self.match

    def print_details(self):
        print('Name: '+self.__name+', Team Name: ' +self.__team)
        print('Team Role:',self.role)
        print('Total Goal:',self.goal,'Total Played:',self.match)
        print('Goal Ratio:',self.ratio)
        self.player_earning = (self.goal*1000)+(self.match*10)
        print('Match Earning: '+str(self.player_earning)+'K')

class Manager(Football):
    def __init__(self,team_name, name, role,win):
        super().__init__(team_name, name, role)
        self.__team = team_name
        self.__name = name
        self.win = win

    def print_details(self):
        print('Name: '+self.__name+', Team Name: ' +self.__team)
        print('Team Role:',self.role)
        print('Total Win:',self.win)
        self.manager_earning = self.win*1000
        print('Match Earning: '+str(self.manager_earning)+'K')
```

```
player_one = Player('Juventus', 'Ronaldo', 'Striker', 25, 32)
player_one.calculate_ratio()
player_one.print_details()
print('----------------------------------------')
manager_one = Manager('Real Madrid', 'Zidane', 'Manager', 25)
manager_one.print_details()
```

```
Name: Ronaldo, Team Name: Juventus
Team Role: Striker
Total Goal: 25 Total Played: 32
Goal Ratio: 0.78125
Match Earning: 25320K
----------------------------------------
Name: Zidane, Team Name: Real Madrid
Team Role: Manager
Total Win: 25
Match Earning: 25000K
```

In [ ]: