Lab 04 – Class Variables and Data Hiding .

Task 01:

Define what do you understand by classes, objects and functionalities.

Class:

Classes are essentially a template to create your objects.

Object:

An object is a collection of data and associated behaviors.

Task 02:

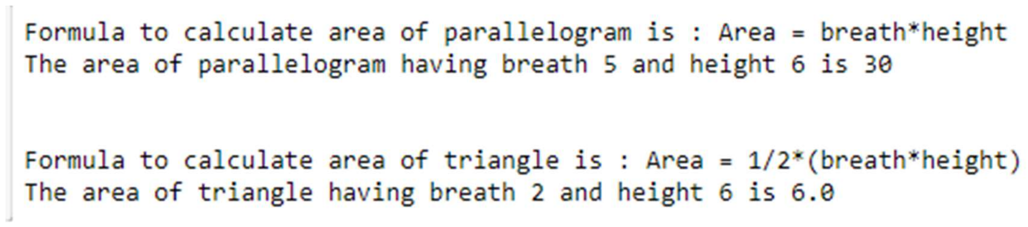
Shape classes and its subclasses and its parameters

Code:

```
class Shape:
    def different_shape(self):
        print("Shapes")


class Parallelogram(Shape):
    def __init__ (self, breath, height):
        self.breath = breath
        self.height = height


    def calculate(self):
        a = self.breath*self.height
        print("Formula to calculate area of parallelogram is : Area = breath*height")
        print(f"The area of parallelogram having breath {self.breath} and height {self.height} is {a}")


shape1 = Parallelogram(5,6)
shape1.calculate()


print("\n")
```

```
class Triangle(Shape):
    def __init__ (self, breath, height):
        self.breath = breath
        self.height = height

    def calculate_area(self):
        a = 1/2*(self.breath*self.height)
        print("Formula to calculate area of triangle is : Area = 1/2*(breath*height)")
        print(f"The area of triangle having breath {self.breath} and height {self.height} is {a}")

shape2 = Triangle(2,6)
shape2.calculate_area()
```

Output:



```
Formula to calculate area of parallelogram is : Area = breath*height
The area of parallelogram having breath 5 and height 6 is 30


Formula to calculate area of triangle is : Area = 1/2*(breath*height)
The area of triangle having breath 2 and height 6 is 6.0
```

Task 03:

Create courses in CS and SE under the class of CS Program. Make methods that can add courses in semester and return it.

Code:

```
class CS_Program:
  def __init__(self, course_1, course_2, course_3, course_4, course_5, course_6):
    self.course_1 = course_1
    self.course_2 = course_2
    self.course_3 = course_3
    self.course_4 = course_4
    self.course_5 = course_5
    self.course_6 = course_6

  def getCourses(self):
    print("Course Name and Course No : ")
    print(f"CS-122 : {self.course_1}")
    print(f"CS-123 : {self.course_2}")
    print(f"HS-121 : {self.course_3}")
    print(f"HS-122 : {self.course_4}")
    print(f"HS-151 : {self.course_5}")
    print(f"CS-121 : {self.course_6}")

  def se_program(self):
    print("Software Engineering")

  def cs_program(self):
    print("Computer Science")

CS = CS_Program("Discrete Structures","Digital Logic Design","Communication Skills","Pakistan
Studies","Financial Accounting",
```

"Object Oriented Programming 3.1")

SE = CS_Program("Discrete Structures","Digital Logic Design","Communication Skills","Pakistan Studies","Financial Accounting",

"Object Oriented Programming 3.1")

CS.cs_program()

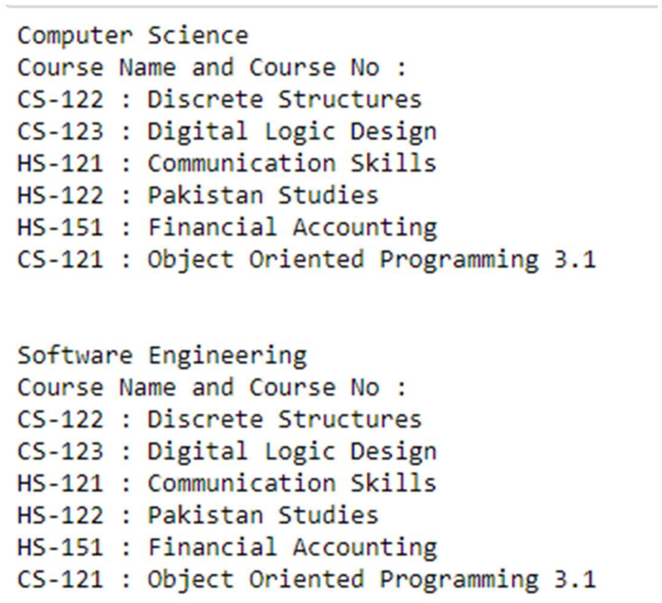
CS.getCourses()

print("\n")

SE.se_program()

SE.getCourses()

Output:



```
Computer Science
Course Name and Course No :
CS-122 : Discrete Structures
CS-123 : Digital Logic Design
HS-121 : Communication Skills
HS-122 : Pakistan Studies
HS-151 : Financial Accounting
CS-121 : Object Oriented Programming 3.1


Software Engineering
Course Name and Course No :
CS-122 : Discrete Structures
CS-123 : Digital Logic Design
HS-121 : Communication Skills
HS-122 : Pakistan Studies
HS-151 : Financial Accounting
CS-121 : Object Oriented Programming 3.1
```

Task 04:

Create a bike class and its components in light with the concept of Object Oriented. Later create multiple bikes with different attributes based on customer requirements.

Code:

```
class Bike:
  def __init__(self, bike_type, design, frame, wheel, front_fork, rear_fork, price):
    self.bike_type = bike_type
    self.design = design
    self.frame = frame
    self.wheel = wheel
    self.front_fork = front_fork
    self.rear_fork = rear_fork
    self.price = price


  def detail(self):
    print(f"Bike Type : {self.bike_type}")
    print(f"Design : {self.design}")
    print(f"Frame : {self.frame}")
    print(f"Wheel : {self.wheel}")
    print(f"Front fork : {self.front_fork}")
    print(f"Rear fork : {self.rear_fork}")
    print(f"Bike Type : {self.bike_type}")
    print(f"Price : {self.price}")
    print("\n")


bike1 = Bike("Road Bike", "Off-Road riding", "Very light frame", "Narrow", "No Suspension", "No
Suspension", "12000")
bike1.detail()
```

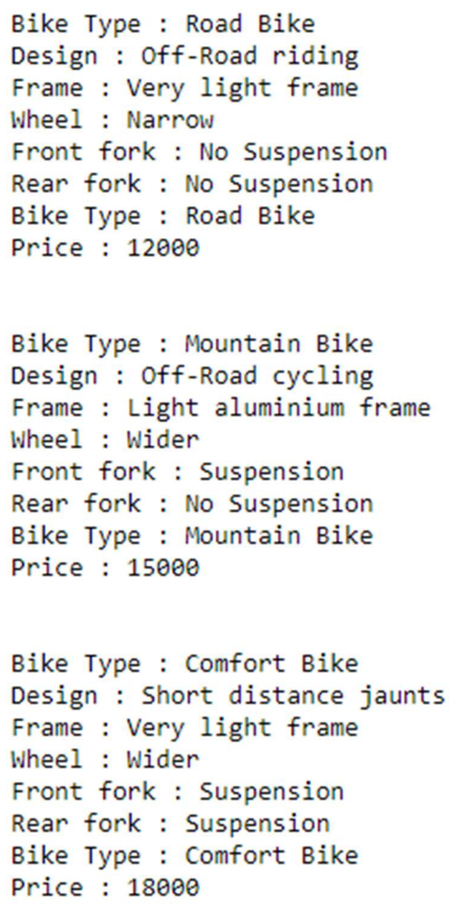
bike2 = Bike("Mountain Bike", "Off-Road cycling", "Light aluminium frame", "Wider", "Suspension", "No Suspension", "15000")

bike2.detail()

bike3 = Bike("Comfort Bike", "Short distance jaunts", "Very light frame", "Wider", "Suspension", "Suspension", "18000")

bike3.detail()

Output:



```
Bike Type : Road Bike
Design : Off-Road riding
Frame : Very light frame
Wheel : Narrow
Front fork : No Suspension
Rear fork : No Suspension
Bike Type : Road Bike
Price : 12000


Bike Type : Mountain Bike
Design : Off-Road cycling
Frame : Light aluminium frame
Wheel : Wider
Front fork : Suspension
Rear fork : No Suspension
Bike Type : Mountain Bike
Price : 15000


Bike Type : Comfort Bike
Design : Short distance jaunts
Frame : Very light frame
Wheel : Wider
Front fork : Suspension
Rear fork : Suspension
Bike Type : Comfort Bike
Price : 18000
```