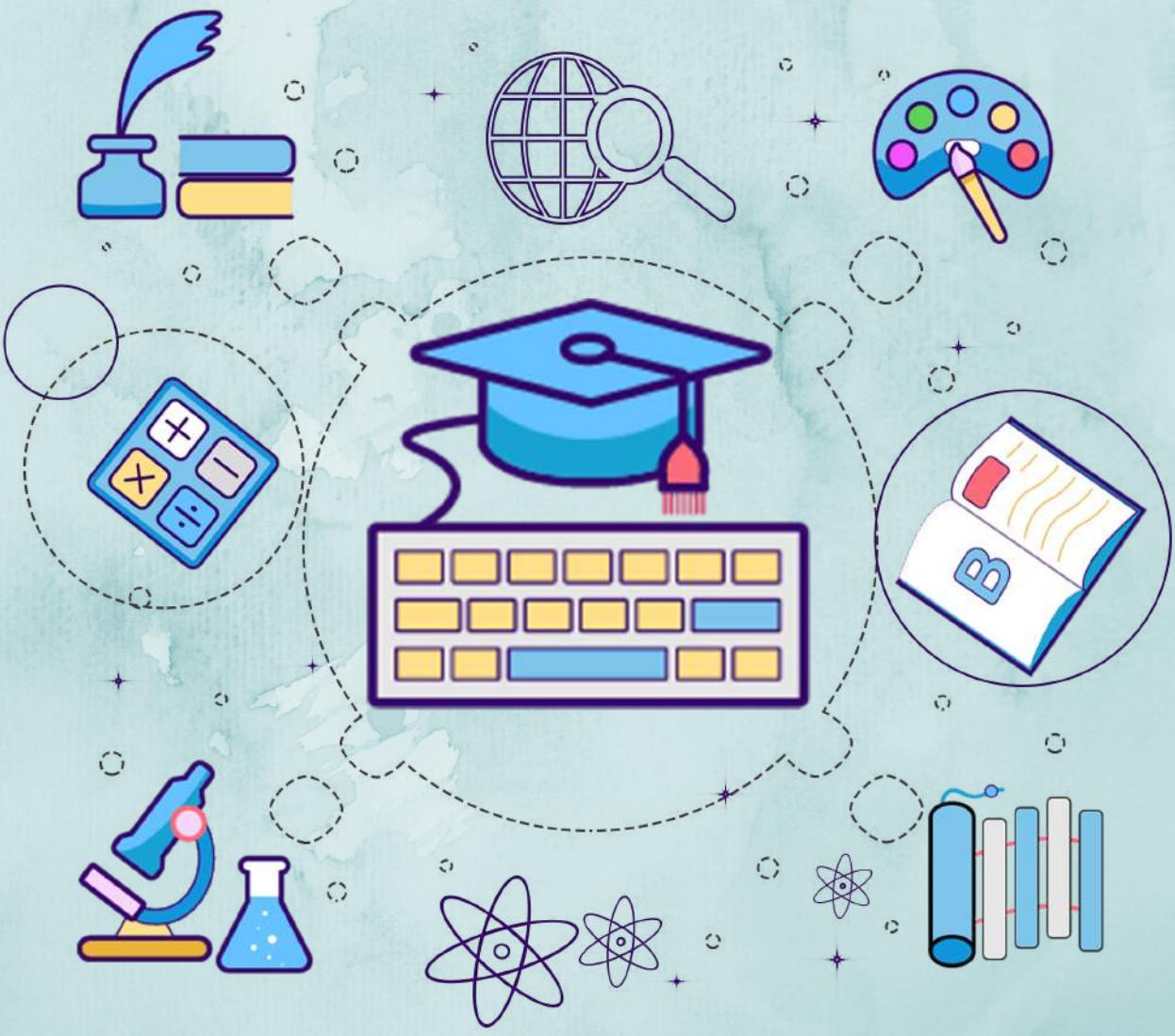
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

KeralaNotes



SYLLABUS | STUDY MATERIALS | TEXTBOOK

PDF | SOLVED QUESTION PAPERS

www.keralanotes.com

KeralaNotes

KTU STUDY MATERIALS

PROGRAMMING IN PYTHON

CST 362

Module 4

Related Link :

- KTU S6 CSE NOTES 2019 SCHEME
- KTU S6 SYLLABUS CSE COMPUTER SCIENCE
- KTU PREVIOUS QUESTION BANK S6 CSE SOLVED
- KTU CSE TEXTBOOKS S6 B.TECH PDF DOWNLOAD
- KTU S6 CSE NOTES | SYLLABUS | QBANK | TEXTBOOKS DOWNLOAD

TUTORIAL QUESTIONS

MODULE-IV

1. Create class Arith to do arithmetic operation. It contains a member function read() to read the two numbers and add() method to find the sum. You can add more methods to the class to incorporate more functionality.

Ans:

```
class Arith:
    def read(self):
        self.x=int(input("enter first number.."))
        self.y=int(input("enter second number..."))
    def add(self):
        print("sum=",self.x+self.y)
#creating an object
A=Arith()
#calling the methods
A.read()
A.add()
```

2. Create a class Rectangle .A constructor is used to initialize the object values. Member function area() to compute the area of the rectangle (university question).

Ans:

```
class Rectangle:
```

```
    def __init__(self,length=0,breadth=0):
        self.length=length
        self.breadth=breadth
    def area(self):
        print("area=",self.length*self.breadth)
R1=Rectangle(10,20)
R1.area()
R2=Rectangle(12,13)
R2.area()
```

3. Create a class car with attributes model, year and price and a method cost() for displaying the prize. Create two instance of the class and call the method for each instance.(university question)

```
class Car:
    def __init__(self,model,year,prize):
        self.model=model
        self.year=year
        self.prize=prize
    def cost(self):
        print("Prize of the car=",self.prize)
C1=Car("Maruti",2004,200000)
C2=Car("Ford",2014,5000000)
C1.cost()
C2.cost()
```

4. Create a class student with attribute name and roll number and a method dataprint() for displaying the same. Create two instance of the class and call the method for each instance.(university question)

```
class Student:
    def __init__(self,name,rno):
        self.name=name
        self.rno=rno
    def dataprint(self):
        print("Name=",self.name)
        print("Rno=",self.rno)
s1=Student("devi",101)
s2=Student("anjana",102)
s1.dataprint()
s2.dataprint()
```

5. Create a class Person with attributes name, age salary and a method display() for showing the details. Create two instances of the class and call the method for each instance.

```
class Person:
    def __init__(self,name,age,salary):
        self.name=name
        self.age=age
        self.salary=salary
    def display(self):
        print("Name=",self.name)
```

```
        print("Age=",self.age)
        print("Salary=",self.salary)
s1=Person("devi",30,10100)
s2=Person("anjana",35,10200)
s1.display()
s2.display()
```

6. Define a class Mobile to store the details of a Mobile (company, model,price) with the following methods.

    a) set_details()- to set the values to the data attributes

    b)display_details()-to display the data attribute values

Create an object of the class and invoke methods. (university question)

```
class Mobile:
    def set_details(self):
        self.company=input("enter compnay name...")
        self.model=input("enter model name..")
        self.price=float(input("enter price.."))
    def display_details(self):
        print("Company Name=",self.company)
        print("Model=",self.model)
        print("Price=",self.price)
M=Mobile()
M.set_details()
```

M.display_details();

7. Define a class in Python to store the details of students(rollno, mark1,mark2) with the following methods

   readData()- to assign values to class attributes

   computeTotal()-to find the total marks

   printDetails()- to print the attribute values and total marks.

Create an object of this class and invoke the methods. (Univesrsity question)

```
class Student:
    def readData(self):
        self.rollno=int(input("enter roll number..."))
        self.mark1=int(input("enter mark1.."))
        self.mark2=int(input("enter mark2.."))
    def computeTotal(self):
        self.total=self.mark1+self.mark2
    def printDetails(self):
        print("roll number-->",self.rollno)
        print("Mark1-------->",self.mark1)
        print("Mark2-------->",self.mark2)
        print("Total Marks---",self.total)
S=Student()
S.readData()
S.computeTotal()
```

```
S.printDetails()
```

8. Define a class in Python to store the details of book(title,author,cost) with the following methods

   get_details()- to assign values to class attributes

   print_details()- to display the attribute values

   Create an object of this class and invoke the methods. (University question)

```
class Book:
    def get_details(self):
        self.title=input("enter book title...")
        self.auth=input("enter author..")
        self.cost=int(input("enter cost.."))
    def print_details(self):
        print("Book Tile-->",self.title)
        print("Author-------->",self.auth)
        print("Cost-------->",self.cost)
B=Book()
B.get_details()
B.print_details()
```

9. How can a class be instantiated in Python? Write a Python program to express the instances as return values to define a class RECTANGLE with parameters height,

width, corner_x, and corner_y and member functions to find center, area, and perimeter of an instance.

```
class RECTANGLE:
    def read(self):
        self.height=int(input("enter height of rectangle.."))
        self.width=int(input("enter width of rectangle.."))
        self.corner_x=int(input("enter right corner x..."))
        self.corner_y=int(input("enter right corner y..."))
    def center(self):
        self.corner_x1=self.corner_x+(self.width/2)
        self.corner_y1=self.corner_y+(self.height/2)
        print("Center=({},{})".format(self.corner_x1,self.corner_y1))
    def area(self):
        print("Area=",self.height*self.width)
    def perimeter(self):
        print("Perimeter=",2*(self.height+self.width))

#creating an object
R=RECTANGLE()
#calling the methods
R.read()
R.center()
```

```
R.area()
```

10. Write Python program to create a class called as Complex and implement __add__( ) method to add two complex numbers. Display the result by overloading the + Operator.

```
class Complex:
    def __init__(self,a=0,b=0):
        self.a=a
        self.b=b
    def display(self):
        if self.b>0:
            print("complex number is ",self.a, "+",self.b, "j")
        else:
            print("complex number is ",self.a, self.b, "j")
    def __add__(self,other):
        r=self.a+other.a
        i=self.b+other.b
        return Complex(r,i)
c1=Complex(2,-3)
c2=Complex(3,4)
c3=c1+c2
c1.display()
c2.display()
```

```
c3.display()
```