Aarya Arban

S11-07

Assignment No.7 – Classes and Objects

```
class MyClass:
x = 5
p1 = MyClass()
print(p1.x)
```

Output:
5

```
#The __init__() Function
class Person:
def __init__(self, name, age):
self.name = name
self.age = age
p1 = Person("Aarya", 19)
p2 = Person("Aman",4)
p3 = Person("Anant",57)
print(p1.name)
print(p1.age)
print(p2.name)
print(p2.age)
print(p3.name)
print(p3.age)
```

Output:
Aarya
19
Aman
4
Anant
57

```
#The __str__() Function
class Person:
def __init__(self, name, age):
self.name = name
self.age = age
def __str__(self):
return f"{self.name}({self.age})"
```

```
p1 = Person("Aarya", 6)
p2 = Person("Amey",41)
p3 = Person("Astro",25)
print(p1)
print(p2)
print(p3)
```

Output:
Aarya(6)
Amey(41)
Astro(25)

```
#Object Methods
class Person:
def __init__(self, name, age):
self.name = name
self.age = age
def myfunc(self):
print("Hello my name is " + self.name)
p1 = Person("Aarya", 3)
p2 = Person("Anmol",40)
p3 = Person("Anant",50)
p1.myfunc()
p2.myfunc()
p3.myfunc()
```

Output:
Hello my name is Aarya
Hello my name is Anmol
Hello my name is Anantpadmanabhan

```
class Person:
def __init__(mysillyobject, name, age):
mysillyobject.name = name
mysillyobject.age = age
def myfunc(abc):
print("Hello my name is " + abc.name)
p1 = Person("Aarya", 36)
p2 = Person("Soham", 36)
p3 = Person("Samir",40)
p1.myfunc()
print(p1.age)
p2.myfunc()
print(p2.age)
```

```
p3.myfunc()
print(p3.age)
p1.age = 40
p2.age = 30
p3.age = 45
p1.myfunc()
print(p1.age)
p2.myfunc()
print(p2.age)
p3.myfunc()
print(p3.age)
```

Output:

```
Hello my name is Aarya
36
Hello my name is Soham
40
Hello my name is Samir
50
Hello my name is Aarya
40
Hello my name is Soham
30
Hello my name is Samir
45
Hello my name is Aarya
Hello my name is Soham
Hello my name is Samir
```

```
class Bike:
name = ""
gear = 0
bike1 = Bike()
bike1.gear = 11
bike1.name = "Mountain Bike"
print(f"Name: {bike1.name}, Gears: {bike1.gear} ")
```

Output:

```
Name: Mountain Bike, Gears: 11
```

```
class Employee:
employee_id = 0
employee1 = Employee()
employee2 = Employee()
employee1.employeeID = 1001
print(f"Employee ID: {employee1.employeeID}")
employee2.employeeID = 1002
print(f"Employee ID: {employee2.employeeID}")
```

Output:
Employee ID: 1001
Employee ID: 1002

```
class Room:
length = 0.0
breadth = 0.0
def calculate_area(self):
print("Area of Room =", self.length * self.breadth)
study_room = Room()
study_room.length = 42.5
study_room.breadth = 30.8
study_room.calculate_area()
```

Output:
Area of Room = 1309.0

```
# Python code to find student grade
class Student:
def __init__(self):
self.__roll=0
self.__name=""
self.__marks=[]
self.__total=0
self.__per=0
self.__grade=""
self.__result=""
def setStudent(self):
self.__roll=int(input("Enter Roll: "))
self.__name=input("Enter Name: ")
print("Enter marks of 5 subjects: ")
for i in range(5):
self.__marks.append(int(input("Subject "+str(i+1)+": ")))
def calculateTotal(self):
```

```
for x in self.__marks:
self.__total+=x
def calculatePercentage(self):
self.__per=self.__total/5
def calculateGrade(self):
if self.__per>=85:
self.__grade="S"
elif self.__per>=75:
self.__grade="A"
elif self.__per>=65:
self.__grade="B"
elif self.__per>=55:
self.__grade="C"
elif self.__per>=50:
self.__grade="D"
else:
self.__grade="F"
def calculateResult(self):
count=0
for x in self.__marks:
if x>=50:
count+=1
if count==5:
self.__result="PASS"
elif count>=3:
self.__result="COMP."
else:
self.__result="FAIL"
def showStudent(self):
self.calculateTotal()
self.calculatePercentage()
self.calculateGrade()
self.calculateResult()
print(self.__roll,"\t\t",self.__name,"\t\t",self.__total,"\t\t",self.__per,"\t\t",self.
__grade,"\t\t",self.__result)
def main():
#Student object
s=Student()
s.setStudent()
s.showStudent()
if __name__=="__main__":
main()
```

Output:
Enter Roll: 7
Enter Name: Aarya
Enter marks of 5 subjects:
Subject 1: 69
Subject 2: 98
Subject 3: 98
Subject 4: 97
Subject 5: 99
7 Aarya 461 92.2 S PASS

```
# Define a class for Checking prime number
class Check :
def __init__(self,number) :
self.num = number
def isPrime(self) :
for i in range(2, int(num ** (1/2)) + 1) :
if num % i == 0 :
return False
return True
if __name__ == "__main__" :
num = 11
check_prime = Check(num)
print(check_prime.isPrime())
num = 14
check_prime = Check(num)
print(check_prime.isPrime())
```

Output:
True
False