

12-11-2020

(67)

LECTURE - 8
KANAV BANSAL

Factorial program using functions:

i/p - `def fun(n):`

`fact = 1`

`for i in range(1, n+1):`

`fact *= i`

`return fact;`

`Print("factorial of 10 is ", fun(10))`

o/p - factorial of 10 is 3628800

to find
A program for the number of digits:

i/p - `n = int(input("Enter the number:"))`

`count = 0`

`while(n > 0):`

`count = count + 1`

`n = n // 10`

`Print("The number of digits in the no.",
count)`

o/p - Enter the number = 45987

The number of digits in the number: 5

OBJECT-ORIENTED PROGRAM (OOP): (68)

This is a method of structuring a program by bundling related properties and behaviours into individual objects.

The object contains data like the raw or Preprocessed materials at each step on a assembly line, and behaviour like the action each assembly line component performs.

OOP TERMINOLOGY:

CLASS: A user-defined prototype for an object that defines a set of attributes that characterize any object of the class. The attributes are data members (class variable and instance variables) and methods, accessed via dot notation.

CLASS VARIABLE: A variable that is shared by all instances of a class. Class variables

are defined with in a class ⁽⁶⁹⁾ but outside any of the class methods. class variables are not used as frequently as instance variables are.

DATA MEMBER: A class variable or instance variable that holds data associated with a class and its objects.

FUNCTION OVERLOADING: The assignment of more than one behavior to a particular function. The operation performed varies by the types of objects or arguments involved.

INSTANCE VARIABLE: A variable that is defined inside a method and belongs only to the current instance of a class.

INHERITANCE: The transfer of the characteristics of a class to the other classes that are derived from it.

(70)

INSTANCE: An individual object of a certain class. An object 'obj' that belongs to a class circle, for example, is an instance of the class circle.

INSTANTIATION: The creation of an instance of a class.

METHOD: A special kind of function that is defined in a class definition.

OBJECT: A unique instance of a data structure that's defined by its class. An object comprises both data members (class variables and instance variables) and methods.

OPERATOR OVERLOADING: The assignment of more than one function to a particular operator.

class with init method:

(71)

i/p - class person:

```
def __init__(self, name):
```

```
    self.name = name
```

```
def say_hi(self):
```

```
    Print("Hello, my name is", self.name)
```

```
P = Person("PAVANI")
```

```
P.say_hi()
```

o/p - Hello, my name is PAVANI

object method:

i/p - class person:

```
def __init__(self, name, age):
```

```
    self.name = name
```

```
    self.age = age
```

```
def myfunc(self):
```

```
    Print("Hello my name is " + self.name)
```

```
P = person("PAVANI", 24)
```

```
P.myfunc()
```

o/p - Hello my name is PAVANI

To modify object properties: (72)

I/p - 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("PAVANI", 23)
```

```
p1.age = 24
```

```
print(p1.name)
```

```
print(p1.age)
```

O/p - PAVANI

24

1/p - class car:

(73)

```
def __init__(self, b, m, c, d, f):
```

```
    self.brand = b
```

```
    self.model = m
```

```
    self.color = c
```

```
    self.doors = d
```

```
    self.fuel_type = f
```

```
def update_color(self, c):
```

```
    self.color = c
```

```
def update_fuel_type(self, f):
```

```
    if (self.fuel_type == 'petrol'):
```

```
        self.fuel_type = f
```

```
    else:
```

```
        print("Exit")
```

```
h = car("Benz", "320d", "Black", "4", "petrol")
```

```
h.color
```

o/p - Black

```
h = car("Benz", "320d", "Black", "4", "petrol")
```

```
h.doors
```


o/p - "4"

(74)

h = car("Benz", "320d", "Black", "4", "petrol")

h.brand

o/p - Benz

h = car("Benz", "320d", "Black", "4", "petrol")

h.fuel-type

o/p - "Petrol"

h = car("Benz", "320d", "Black", "4", "petrol")

h.model

o/p - "320d"