

Classes & Objects

Content

- Classes & Object concept
- OOPS *
- Syntax * language agnostic (basically for Python)

Class → It is a blueprint (for objects)
eg floor plan of a house/building

Object → Real instance of a class
eg Physical house/building
(one class can have multiple objects)

Class {
Attributes (used to define data)
Methods (to define functionality)

class Car {
name
color
category
mileage
⋮
}

attribute
property

objects

Car:
name → Audi A4
color → Black
category → sedan
mileage → 8 kmpl
⋮

Car:
name → BMW
color → Blue
category → SUV
mileage → 10
⋮

method/ functionality {
 drive() {}
 AC() {}
 music() {}
 :
 }

drive() {} drive() {}
 AC() {} AC() {}
 music() {} music() {}
 :
 :
 same functionalities in all objects

class Student {
 attributes {
 String name
 int id
 :
 }
 methods {
 study() {}
 exam() {}
 bunk() {}
 :
 }
}

Student S1 = new Student()
 [#2368] ✓
 object reference
 of Student class

name = "Easwar"
 id = 32

[#2368]
 memory address

S1.name = "Easwar"

S1.id = 32

↳ dot is used to access attributes & methods

S1.study()

Student S2 = new Student()

S2.name = "Anil"

S2.id = 21

Student S3; [→ null]

print(S3.name)

↳ Null Pointer Exception error

name = "Anil"
 id = 21

[#1234]

Student S4 = S2; → shallow copy
 [#1234]

print(S2.name) → "Anil"

S4.name = "Rahul"

print(S2.name) → "Rahul"

```
int a = 2  
b = a  
b = b + 2  
print(a) → 2
```

Student S5 = S4
S2.name = "Sagar"
print(S5.name) → "Sagar"

Student S6 = new Student()
S6.name = "Sanjay"
S6.id = 87

→ name = "Sanjay"
id = 87 (# 6791)

Student S7 = new Student()
S7.name = S6.name
S7.id = S6.id

→ name = "Sanjay"
id = 87 (# 6911)

print(S7.name) → "Sanjay"
S6.name = "Kiran"

print(S7.name) → "Sanjay"
print(S6.name) → "Kiran"

Deep copy

Question 1

Create a class Rectangle that support following methods:

1. Find the area of rectangle
2. Check if rectangle is square or not?

Class Rectangle {

int length, breadth;

Rectangle(x, y) {

length = x, breadth = y

} int area() {

return length * breadth

}

bool isSquare() {

return (length == breadth)

}

}

Rectangle r = new Rectangle()

r.length = 4

r.breadth = 6

l = 4

b = 6

print(r.area()) → 24

Constructor →

Special method used to initialize the attributes of a class at a time of object creation.



Rectangle r2 = new Rectangle(4, 6)

print(r2.area()) → 24

1. Name is same as class name

2. No return type

Question 2

Given N rectangles with length & breadth in $A[]$ & $B[]$.

$(A[i], B[i]) \rightarrow i^{\text{th}}$ rectangle

find the sum of area of rectangles which are not square using Rectangle class.

eg $A = [2, 5, 3, 6, 2]$
 $B = [4, 5, 1, 6, 2]$

$$\text{ans} = (2 \times 4) + (3 \times 1) \\ = 8 + 3 = 11$$

$n = A.length$

$sum = 0$

for ($i=0; i < n; i++$) {

Rectangle $r = \text{new Rectangle}(A[i], B[i])$

if ($! r.isSquare()$) {

$sum += r.area()$

}

}

print(sum)

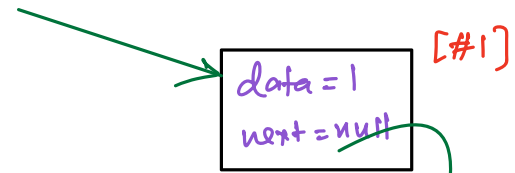
Using Class Improves

1. Readability
2. Reusability

Object Reference inside a class

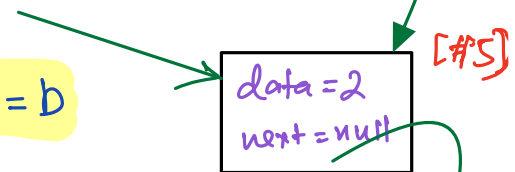
```
class Node {  
    int data;  
    Node next; // obj reference  
  
    Node(int n) {  
        data = n  
        next = null  
    }  
}
```

Node a = new Node(1)



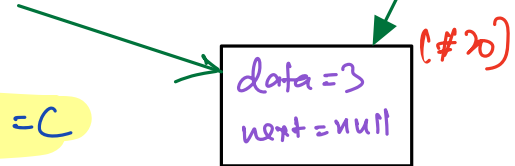
Node b = new Node(2)

a.next = b



Node c = new Node(3);

b.next = c



Linked List