Static variable creation :- (b) Inside class outside of the constructor

MAIN MEMORY

STACK MEMORY

HEAP MEMORY

For this class python will create an object internally.

```
class Student:
    college = 'NIT college'
    branch = 'CSE'

    def __init__(self, name, age, roll):
        self.name = name
        self.age = age
        self.roll = roll

    def printdetails(self):
        print(f'Name is {self.name}')
        print(f'Age is {self.age}')
        print(f'roll is {self.roll}')
        print('-----')

s1 = Student('priyanka', 24, 1024)
s1.printdetails()
s2 = Student('Rahul', 25, 1025)
s2.printdetails()
s3 = Student('zini', 24, 1026)
s3.printdetails()
print(s1.__dict__)
print(Student.__dict__),
```

static variable

Here the output will show that it includes class level variable i.e here college and branch.

Output Screen

```
Name is priyanka
Age is 24
roll is 1024
-----
Name is Rahul
Age is 25
roll is 1025
-----
Name is zini
Age is 24
roll is 1026
-----
{'name': 'priyanka', 'age': 24, 'roll': 1024}
```

* After successfully execution of the function it will vanished from the memory.

* At the time of object creation constructor will execute automatically.

printdetails(self)

self

self

s3

printdetails(self)

self

self

s2

printdetails(self)

self

self

s1

Student

name zini
age 24
roll 1026

name Rahul
age 25
roll 1025

instance variable

name priyanka
age 24
roll 1024

college NIT college
branch CSE

static variable

2022/12/25 00:28

Static variable creation :— (1) Inside class outside of the constructor

For this class python will create an object internally.

static variable

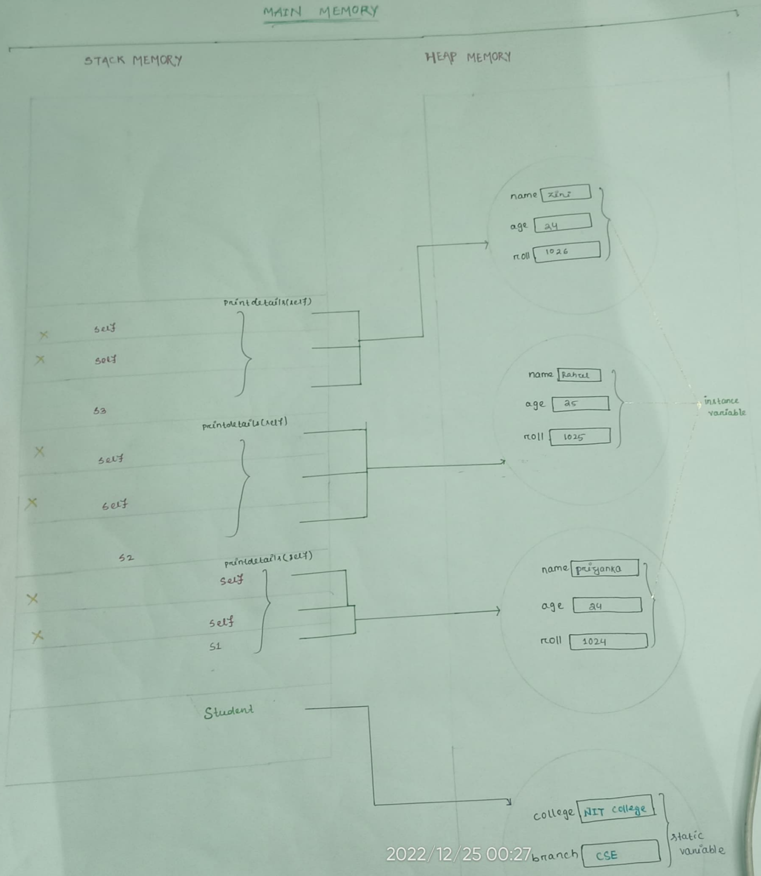
```
class Student:
    college = 'NIT college'
    branch = 'CSE'
    def __init__(self, name, age, roll):
        self.name = name
        self.age = age
        self.roll = roll
    def printdetails(self):
        print(f'Name is {self.name}')
        print(f'Age is {self.age}')
        print(f'roll is {self.roll}')
        print('————')
s1 = Student('priyanka', 24, 1024)
s1.printdetails()
s2 = Student('Rahul', 25, 1025)
s2.printdetails()
s3 = Student('zini', 24, 1026)
s3.printdetails()
print(s1.__dict__)
print(Student.__dict__)
```

Here the output will show that it includes class level variable i.e here college and branch.

* After successfully execution of the function it will vanished from the memory.
* At the time of object creation constructor will execute automatically.

Output screen

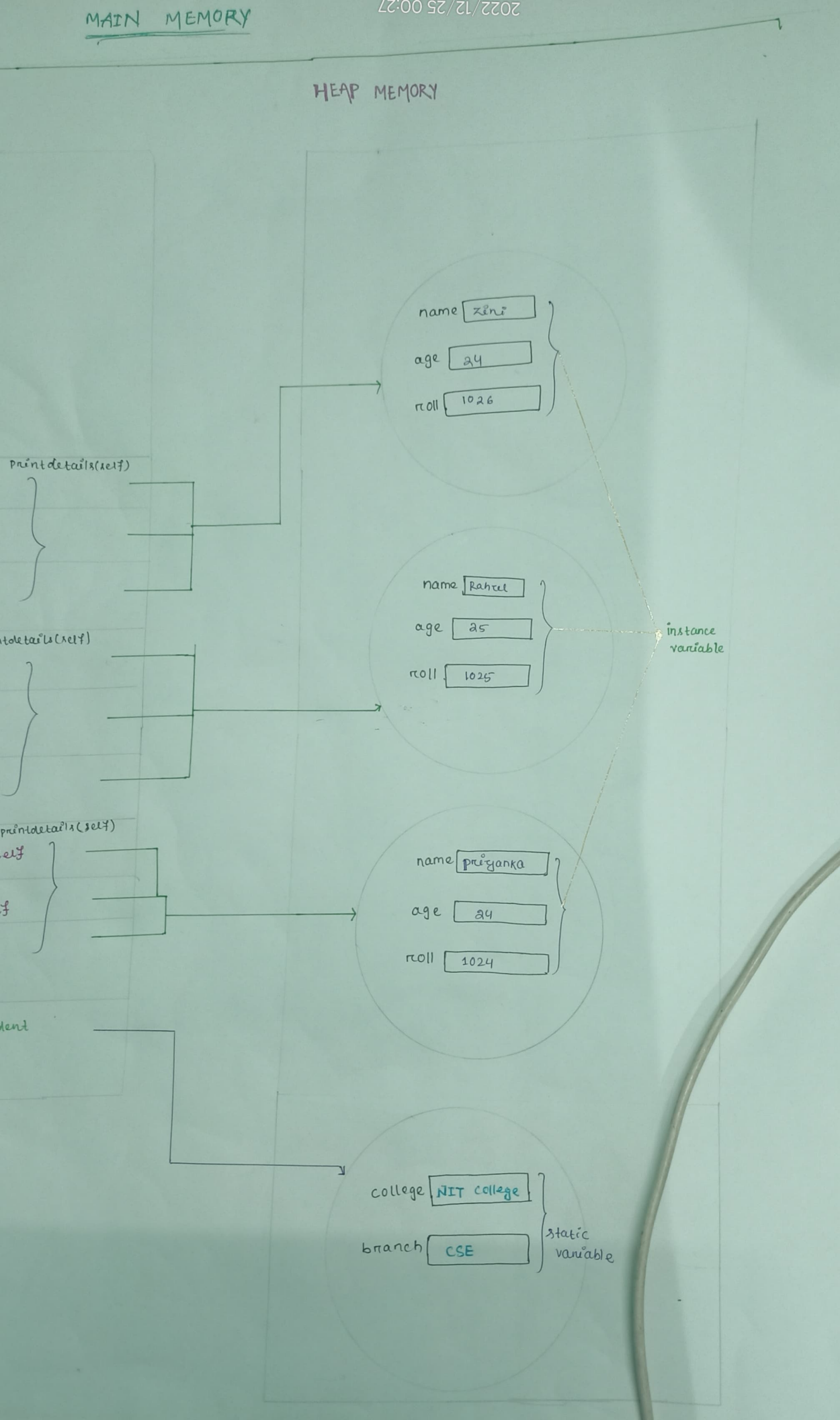
```
is priyanka
is 24
is 1024
— —
s Rahul
s 25
s 1025
—
zini
24
1026
—
ge': 24, 'roll': 1024 }
```



2022/12/25 00:27

MAIN MEMORY

HEAP MEMORY



2022/12/25 00:27

STACK MEMORY

HEAP MEMORY

printdetails(self)

self

self

S3

printdetails(self)

self

self

S2

printdetails(self)

self

self

S1

Student

name

age

roll

name

age

roll

name

age

roll

college

branch

Static variable creation :- (1) Inside class outside

For this class python will create an object internally.

```
class Student:
    college = 'NIT college'
    branch = 'CSE'

    def __init__(self, name, age, roll):
        self.name = name
        self.age = age
        self.roll = roll

    def printdetails(self):
        print(f'Name is {self.name}')
        print(f'Age is {self.age}')
        print(f'roll is {self.roll}')
        print('-----')

s1 = Student('priyanka', 24, 1024)
s1.printdetails()
s2 = Student('Rahul', 25, 1025)
s2.printdetails()
s3 = Student('zini', 24, 1026)
s3.printdetails()
print(s1.__dict__)
print(Student.__dict__)
```

static variable

Here the output will show that it includes class level variable i.e here college and branch.

Output screen

```
Name is priyanka
Age is 24
roll is 1024
----
Name is Rahul
Age is 25
roll is 1025
----
Name is zini
Age is 24
roll is 1026
----
{'name': 'priyanka', 'age': 24, 'roll': 1024}
```

* After successfully execution of the function it will vanished from the memory.

* At the time of object creation constructor will execute automatically.

STACK MEMORY

self

self

s3

self

self

s2

Static variable creation :- (b) Inside class outside of the constructor

```
class Student:
    college = 'NIT college'
    branch = 'CSE'

    def __init__(self, name, age, roll):
        self.name = name
        self.age = age
        self.roll = roll

    def printdetails(self):
        print(f'Name is {self.name}')
        print(f'Age is {self.age}')
        print(f'roll is {self.roll}')
        print('------')

s1 = Student('priyanka', 24, 1024)
s1.printdetails()
s2 = Student('Rahul', 25, 1025)
s2.printdetails()
s3 = Student('zini', 24, 1026)
s3.printdetails()
print(s1.__dict__)
print(Student.__dict__)
```

For this class python will create an object internally.

Here the output will show that it includes class level variable i.e here college and branch.

Output screen

```
Name is priyanka
Age is 24
roll is 1024
--------
Name is Rahul
Age is 25
roll is 1025
--------
Name is zini
Age is 24
roll is 1026
--------
{'name': 'priyanka', 'age': 24, 'roll': 1024}
```

* After successfully execution of the function it will vanished from the memory.

* At the time of object creation constructor will execute automatically.

MAIN MEMORY

STACK MEMORY

HEAP MEMORY