OOP

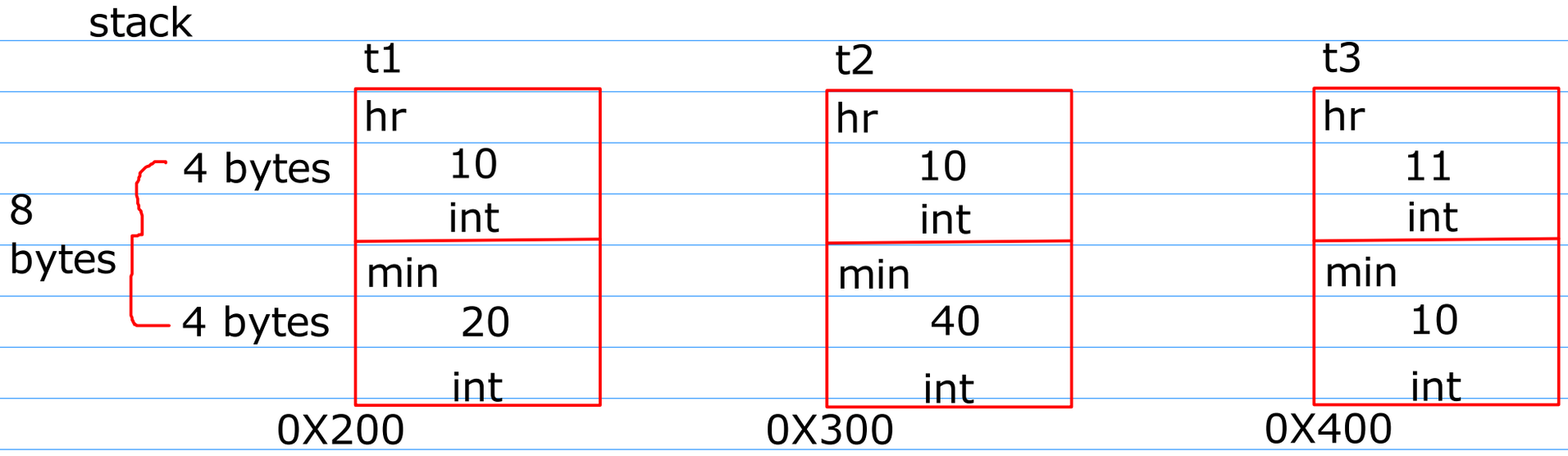
class

- It is a logical entity
- It is also called as blueprint of an object
- class consists of data members and member functions

Object

- A variable of a class type is called as object
- It is a physical entity
- It is also called as instance of the class

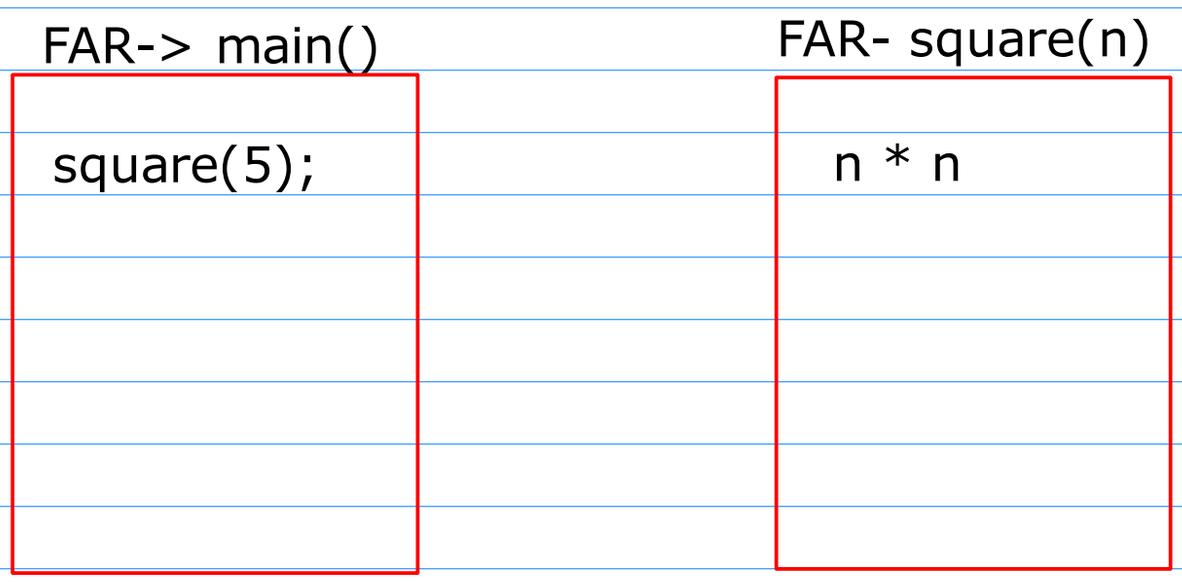
stack



Time t1;

Access Specifier for the class

1. public -> all the public members of the class are accessiable outside the class on class object
2. private -> all the private members are accessiable only within the class and not outside the class
3. protected-> accessiable only within the class and the derived classes



creation of FAR - x
calling the function - y
destruction FAR - z

x+y+z > execution time

#define SIZE 5.5

```
inline int size(){
return 5;
}
```

Name Mangling

namespace
- It is a container used to oraganize the code

Camel Case
total
totalSalaray
calculateTotalSalary()

```
namespace std{

}
```

Pascal Case
Employee
EmployeeAttendance

Object defines 3 things
1. state - The data members of the class represent the state of an object
2. behaviour - The member functions represent the behaviour of an object
3. identity - unique field inside the class will be used as an identity.
If the unique field is not present then address will be considered for its identity

```
class Date{

}
```

```
Date d1;
d1.deposit()

Date d2;
```

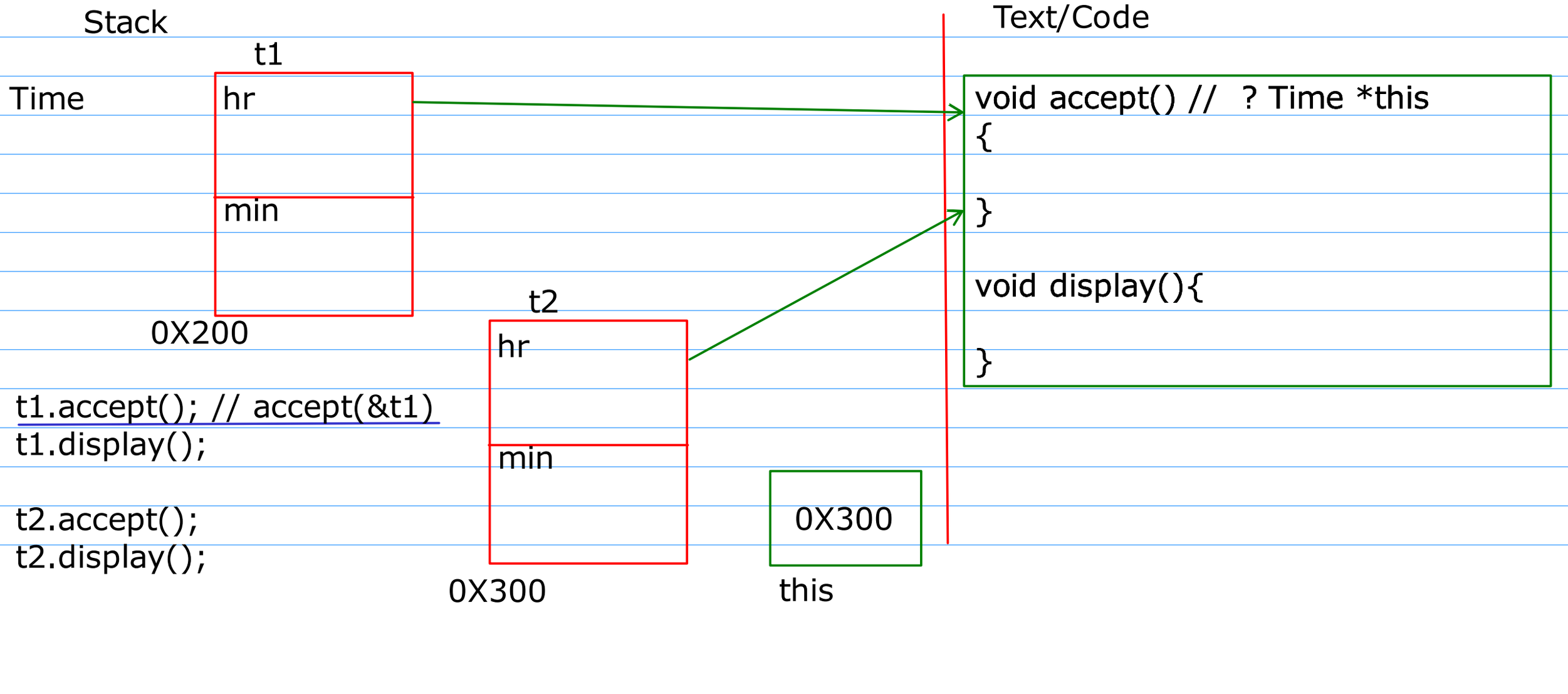
```
class BankAccount{
int accno;
string name;
double balance;

}

BankAccount a1;    // 1001
BankAccount a2;    // 1002
```

d1 - 24/2/2025
d2 - 25/2/2025

Stack Data Heap Text/Code



```
void f1(){
int num = 10;
}

void f2(){
int num = 20;
}

void f3(){
int num = 30;
}

void f4(){
int num = 40;
}
```

```
f1();
f2();
f3();
f4();
```