```
static int menu(){

choice;
}
static void main(){
int choice = menu
}
```

```
int n;
Integer.toBinaryString(n);
```

Datatypes
Primitive (Value types)
1. Boolean
2. Character
3. Integral
4. Floating-Point
NonPrimitive (Reference types)

Packages

com.sunbeam.entity
com.sunbeam.tester

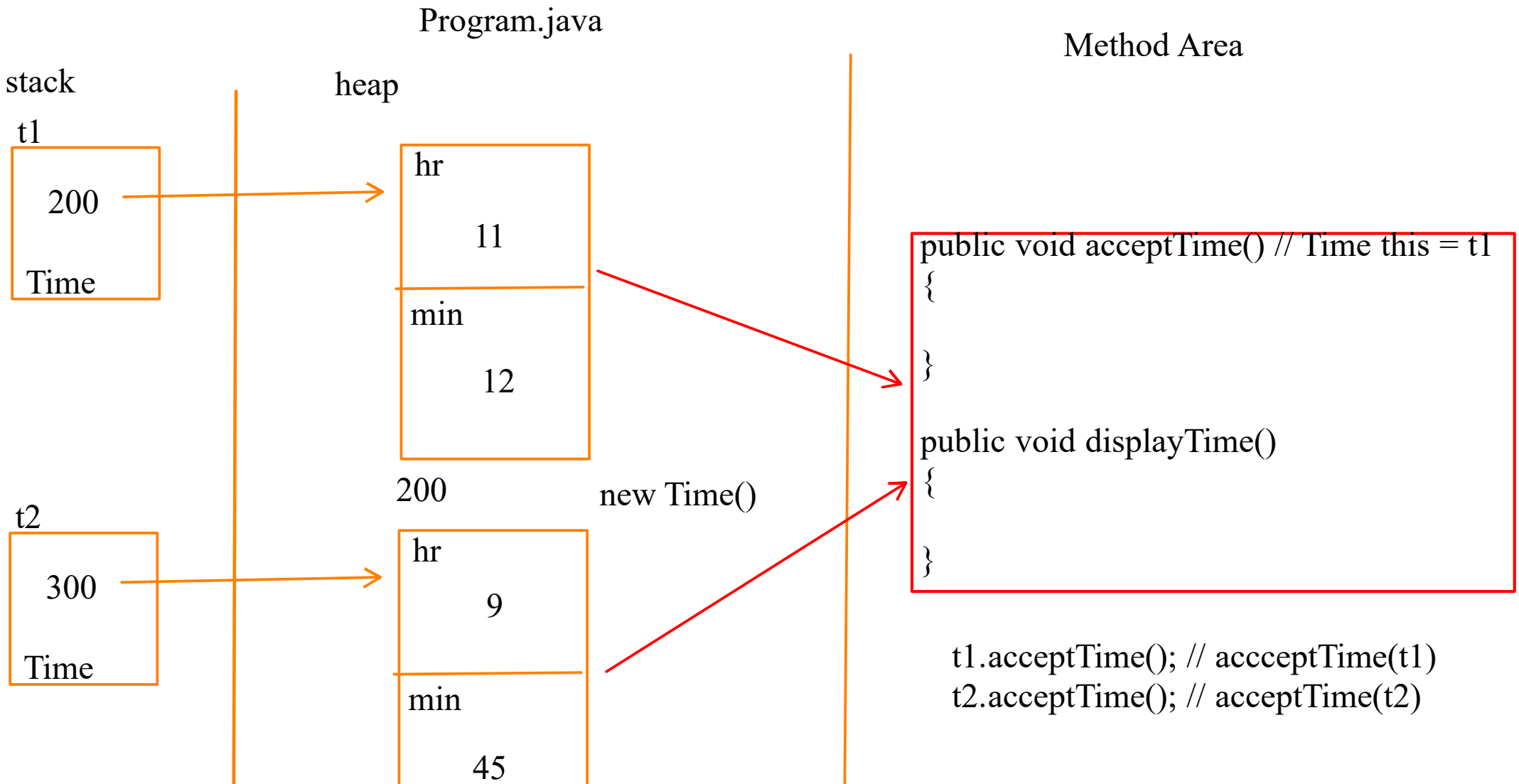
p1

com.sunbeam
Program03.java

com.sunbeam.p1
Time.java
Program02.java

com.sunbeam.p2
Program.java

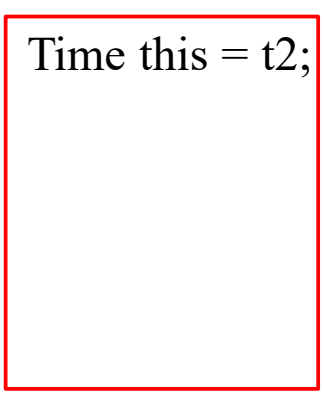
Access Modifiers (Visibility)
private
default
protected
public

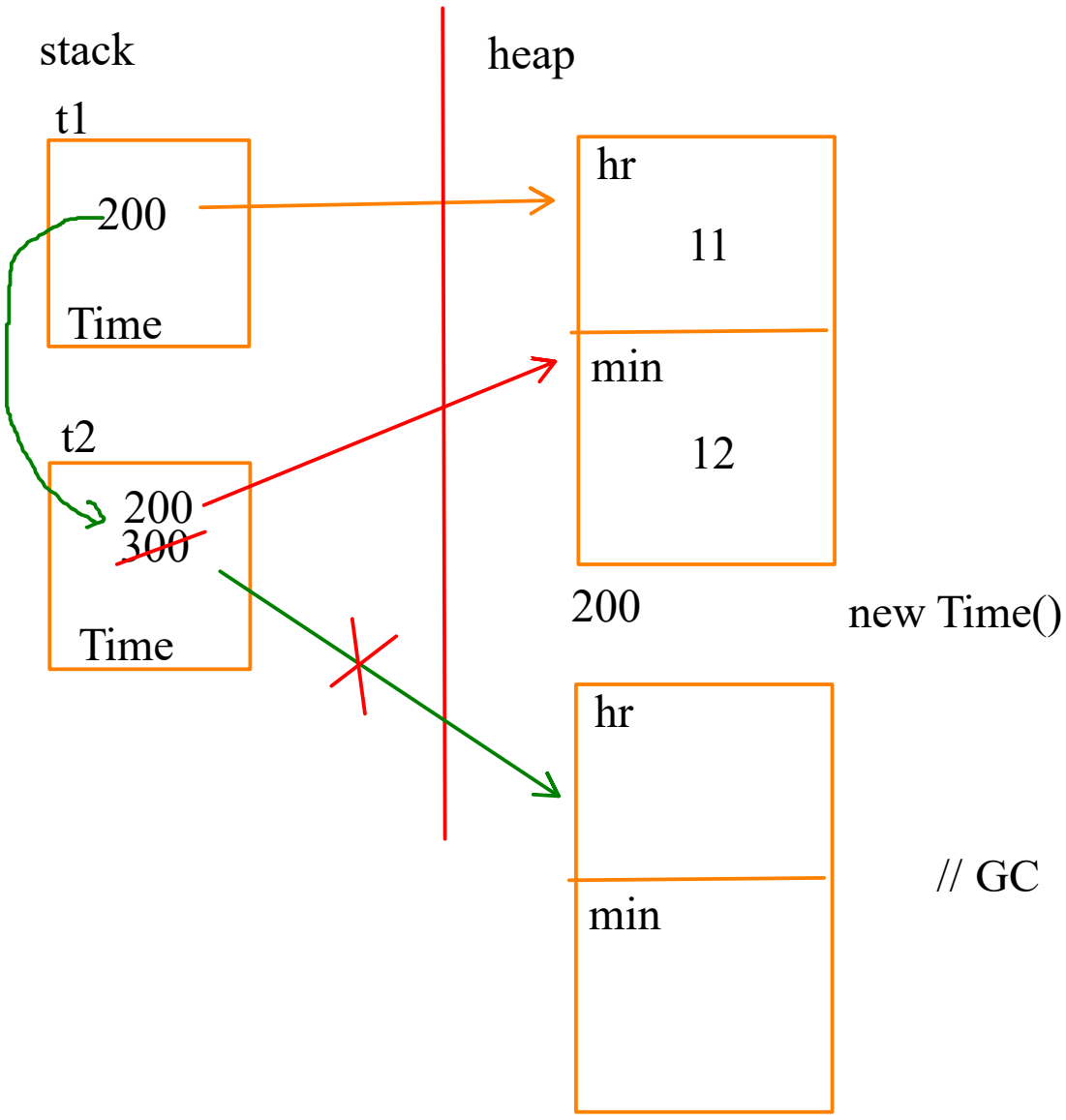



```
// CPP
int main(){
Point p1;
Point &ref = p1;
return 0;
}
```

C
define -> call multiple times

FAR





Time this;

Time t1 = new Time;
this=t1;

Java
C++

300

this reference
- It is a reference that is passed internally to all the non static methods of the class
- It points at the current calling object
- using this is completly optional

Method/Function Overloading

```
- void mutiply(int n1, int n2){
        sysout(n1 * n2)
}
void mutiply(int n1, int n2,int n3){
        sysout(n1 * n2 * n2)
}
```

```
void square(int n){
sysout(n * n);
}
void square(double n){
sysout(n * n);
}
```

```
void division(int n, double d){
sysout(n/d);
}
void division(double n, int d){
sysout(n/d);
}

division (12.50, 2);
```

Method Overloading
- Defining multiple methods with same name but different signature is called as Method overloading
- The signature can be changed in 3 ways
    1. By changing the no of parameters
    2. If no of parameters are same then change the type of parameters
    3. If no and type are same then change order of parameters.
- In method overloading the return types are not considered

Types of Methods
- Constructor
- Setters
- Getters
- Facilitators

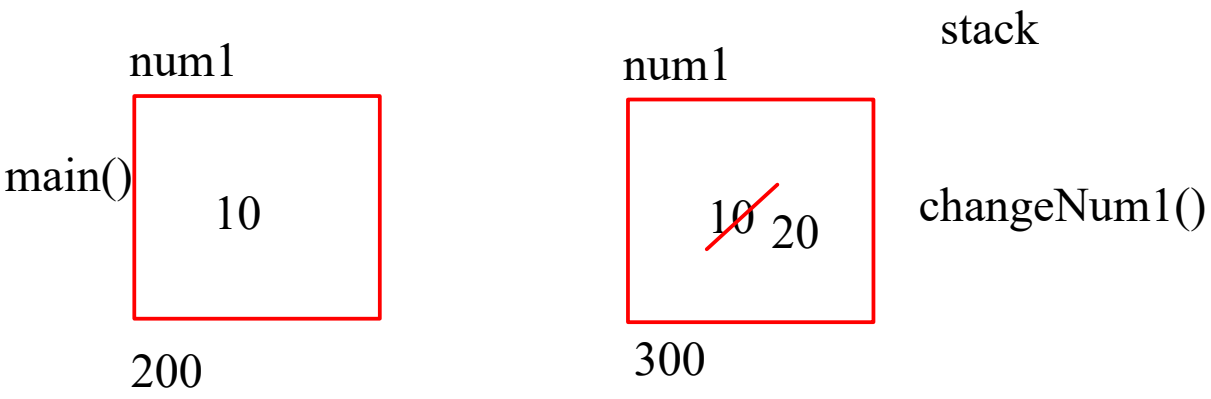
#Constructor
It is special method of a class
- why special
    1. Its name is same as class name
    2. It do not have any return type
    3. It gets called automatically when object is created
- It is used to initialize the state of an object.

#Types of Constructor
- Default/ Parameterless
- Paramaterized

Constructor Chaining
- Calling one constructor from another constructor is called as Constructor Chaining
- for ctor chaining we must use this() statement.
- this() statement should be the first statement in the ctor body.

Initializers
1. Field Initializer
2. Object Initializer
3. Constructor

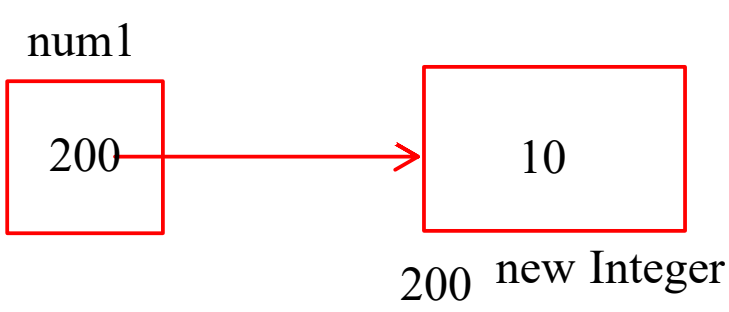
pass by value and reference
- Primitive types are always passed by value
- Non Primitive types are always passed by reference

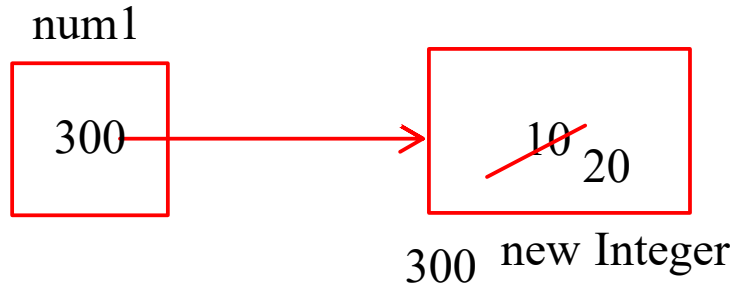



```
main(){
Integer i1 = new Integer(10);
}
```

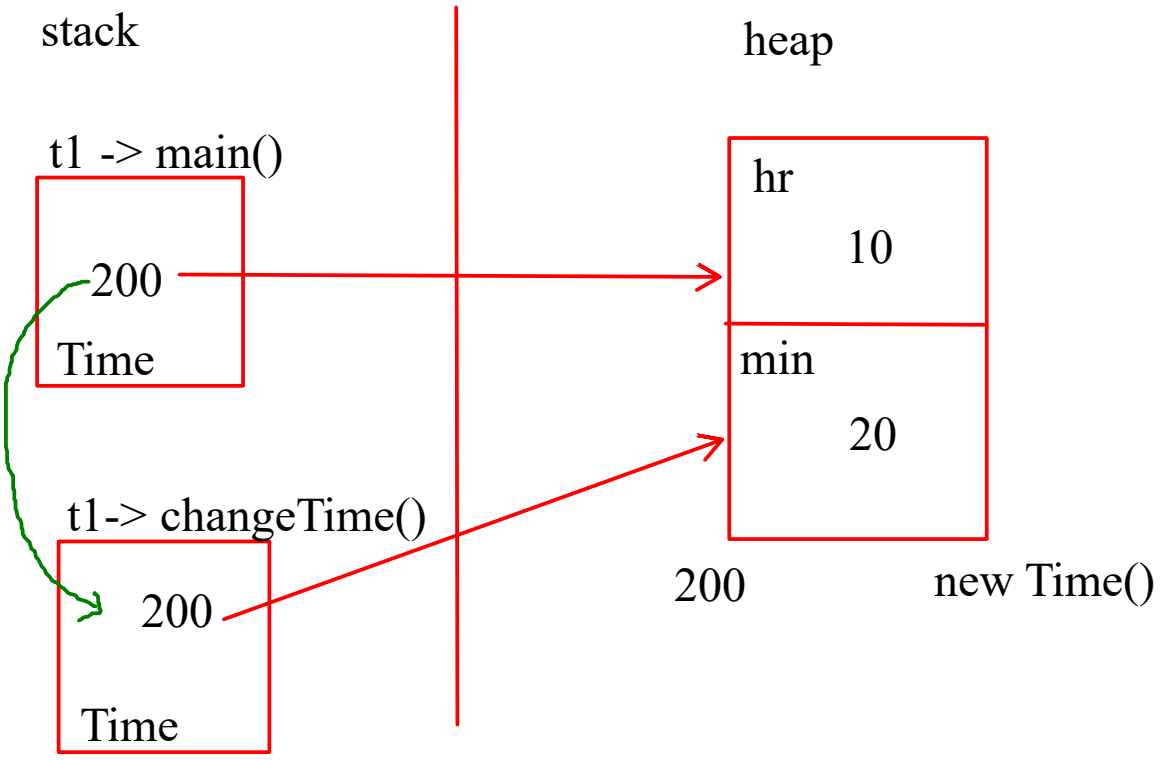
```
changeNum1(Integer num1){

}
```





num1 -> 10 (value)



t1 -> 200

C
// pass by value
// pass by address

CPP
// pass by value
// pass by address
// pass by reference

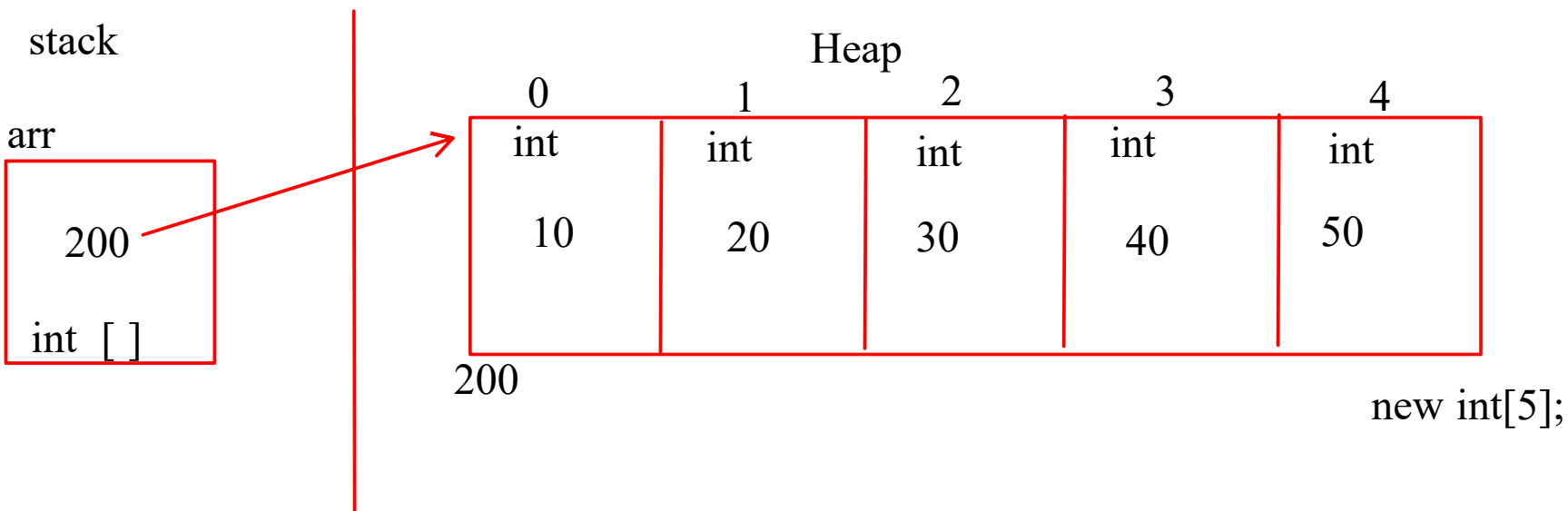
Java
// pass by value
// pass by reference

Lab ->
- Revise the classwork
Solve the assignments

Initializers

this(1,1,2000);

Array
- Array is a referenece type in java
-



//200[0] = 10
arr[0] = 10