OOP -> Java 8 WORA

JRE -> java is considered as portable

Language Fundamentals

Camel Case
Pascal Case
naming convention for packages
naming convention for static and final(constant)

Camel case
- Every first letter of the word should be capital except first word
- salary, totalSalary, calculateTotalSalary
- local variables, fields, method name, parameters

Pascal case
- Every first letter of the word should be capital
- Student , EmployeeException
- class, interface, enum

packages
- every word should be in small case

static and final variables
- It should be in upper case

Class
- It represents a logical entity
- It is also called as blueprint of an object
- It consists of fields and methods

```
class {
// binding the data and code
// fields -> data
// methods -> code
}
```

Object
- It is a physical entity
- It is also called as instance of the class
- Object defines 3 things

1. state
   - Fields of the class represents state of an object
2. behaviour
   - Methods of the class represnts behaviour of an object
3. Identity
   - Uniqe field represent the identity and if unique field does not exist the the address represents the identity

```
int main(){
int inhr;
int inmin;
int outhr
int outmin;
int hr3;
int min3;
int hr4;
int min4;
enter the hrs and mins
// accept
}
```

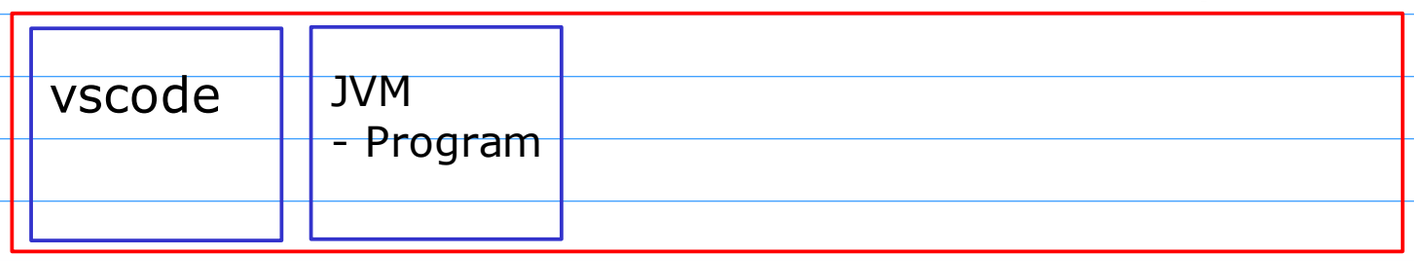
```
class Time{
int hr;
int min;
}

int main(){
Time in;
Time out;
Time t2;
Time t3;

}
```

Datatypes
- It defines 3 things
1. Nature
    - Type of data that can be stored (Number, Alphabets, Words, true)
2. Memory
    - The amount of memory required to store the data
3. Operations
    - Types of operations that can be carried out on that data

Types of Datatype
1. Primitive Type (Value Types)
    a. Boolean
        - boolean -> 1 bit (true and false)
    b. Character
        - char -> 2 bytes
    c. Integeral
        - byte (1 byte), short(2 bytes), int(4 bytes) , long(8 bytes)
    d. Floating-Point
        - float(4 bytes), double(8 bytes)
2. Non Primitive Type (Reference Types)
        - Class
        - Interface
        - Enum
        - Array

Local Variable
Global Variable
Static Varibales

Stack -> Local Variables
Heap -> Dynamic Memory Allocation
Data -> Global / Static
Text/Code -> code(instruction)

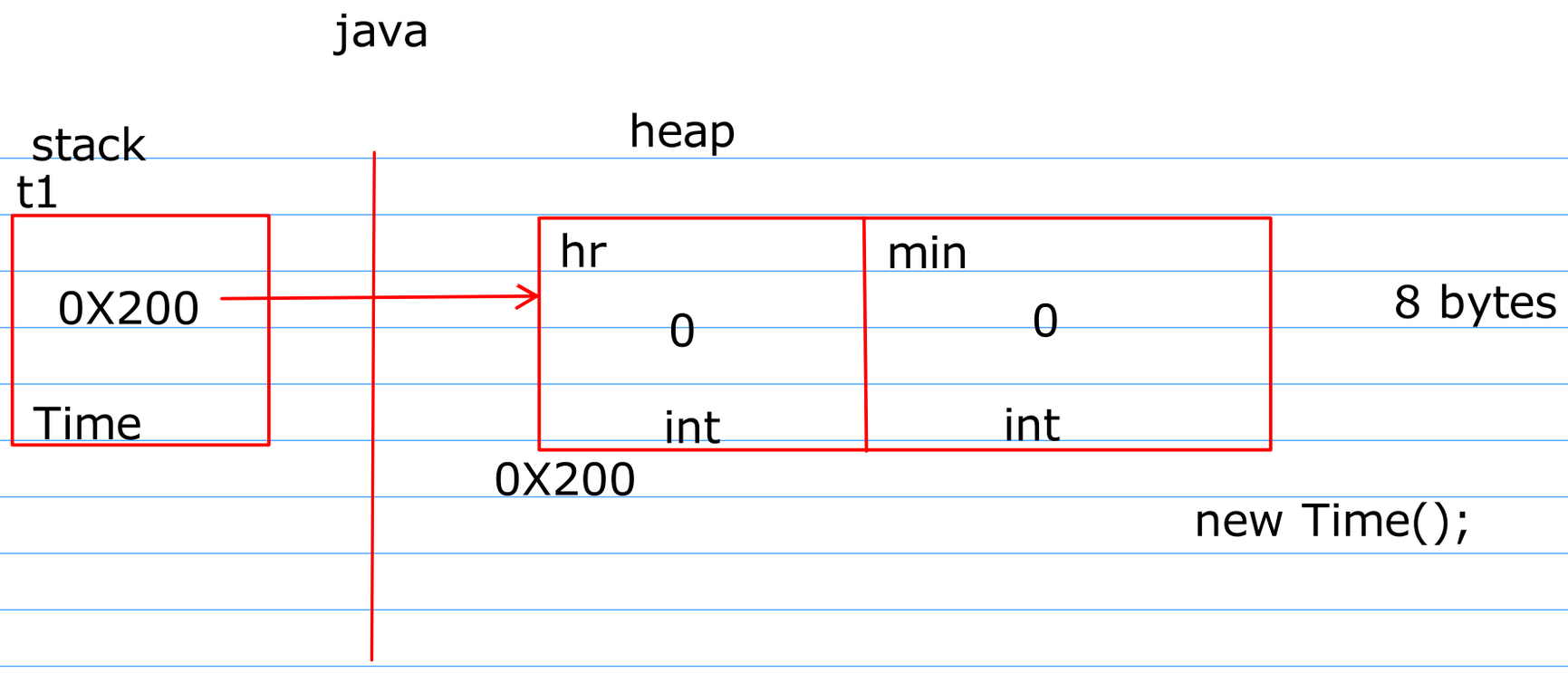


RAM

JRE
- JVM

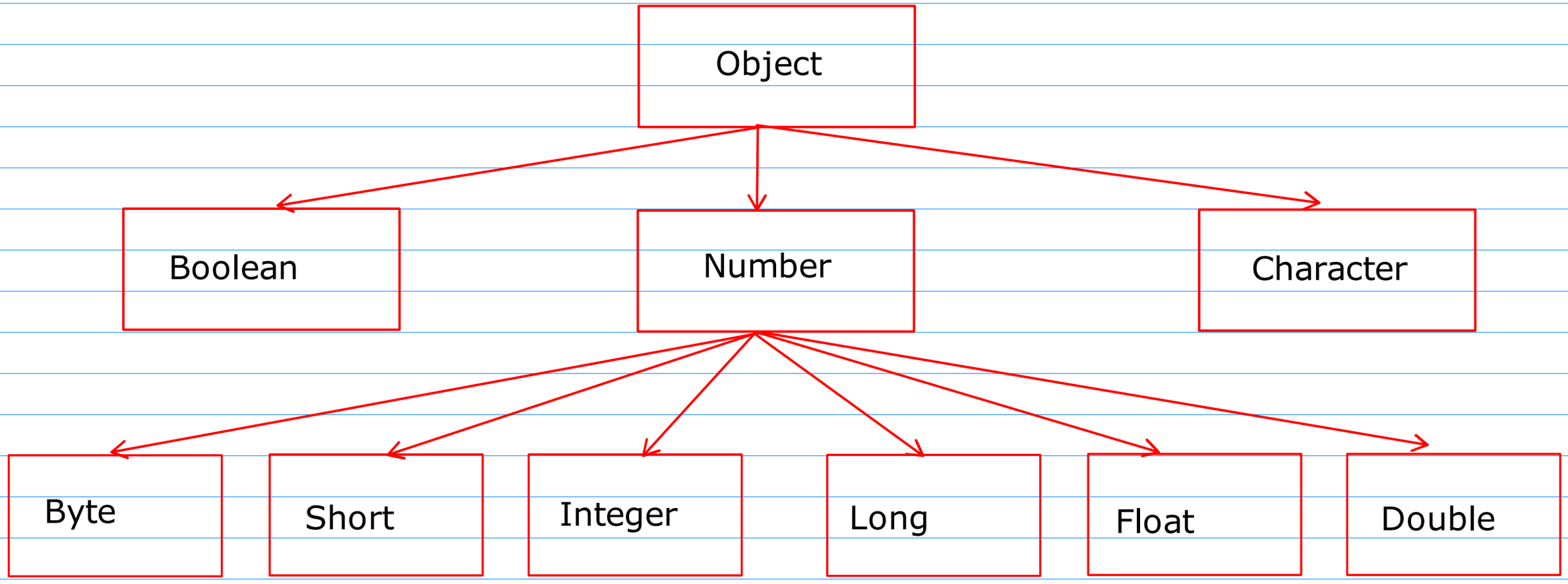
java Program

Memory Areas of JVM
1. Method Area -> static/ methods
2. Java Stack -> Local varaiables
3. Java Heap -> Objects
4. Native Method Stack
5. Pc Register



Wrapper class

- For every primitive type java have provided the classes called as Wrapper class
- these are present in java.lang package



purpose of wrapper class is to use primitive types as objects

Collection

LinkedList<Integer>

Stack<Double>

Queue<Charcter>

compay domain name in reverse order	com.sunbeam.ecommerce.entity
sunbeam.com	com.sunbeam.ecommerce.tester
com.sunbeam	com.sunbeam.ecommerce.utils
tcs.co.in	
in.co.tcs	

Access Modifiers (Visibility)
1. private
2. default
3. protected
4. public

fields
methods

class

Datatypes

Class,Object,Reference

method overloding

add(int,int);
add(int,int,int);
div(int n,double deno);
div(double deno,int n);

name mangling

add_i_i
add_i_i_i
div_i_d
div_d_i

Mangaled Names

Integer i1 = new Integer(10);
i1.intValue();

Integer.parseInt("10");

main(){

}

static

java Program

Program.main()