PW SKILLS

Lecture

OOPs Fundamentals

List of Concepts Involved:

- Object Creation
- Instance variable vs local variable
- Methods with Memory Map(JVM area)
- Method Overloading

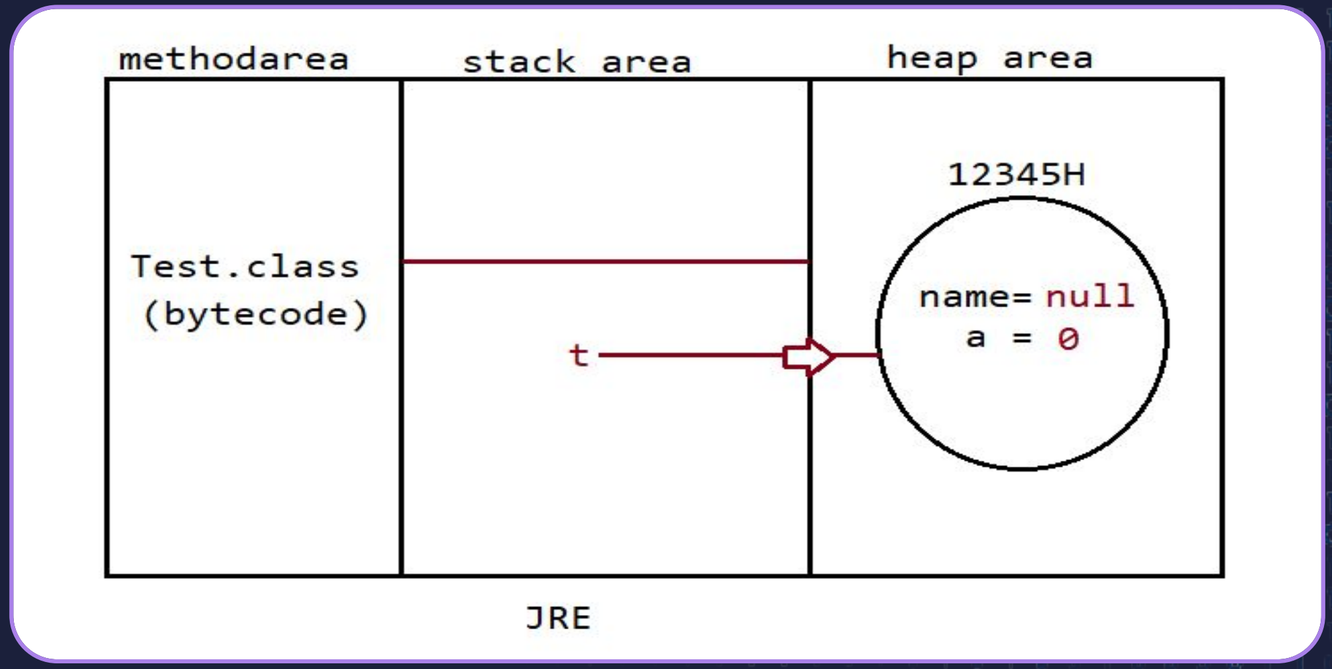
Topics covered Yesterday's Session:

- Pattern Programs

Object Creation

1. We can use the "new" operator to create an object.

2. There is no "delete" operator in java because destruction of useless objects is the responsibility of the garbage collector.



Types of Variables

Based on the behaviour and position of declaration all variables are divided into the following 3 types.

1. Instance variables
2. Static variables
3. Local variables

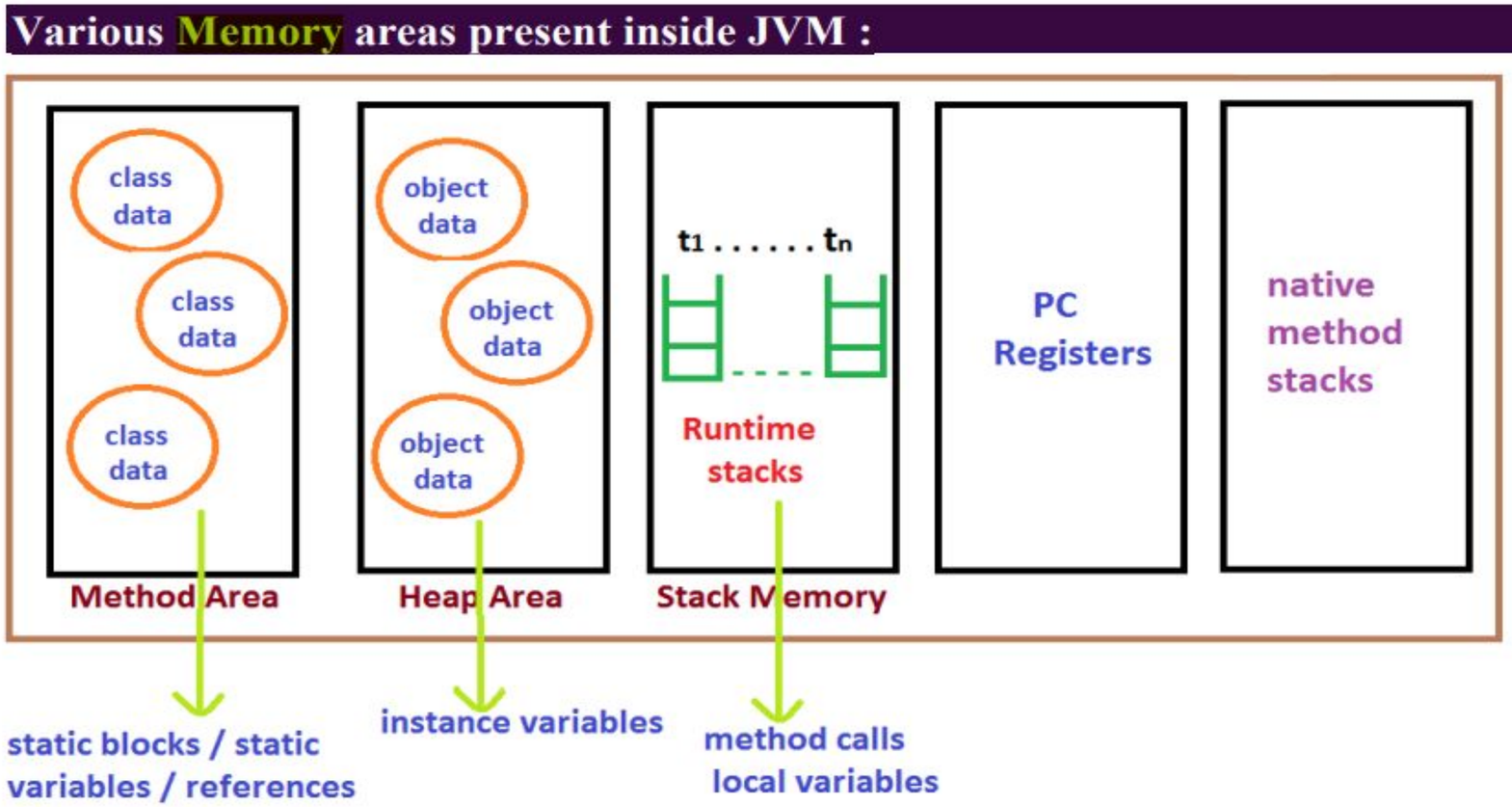
Instance variable

- Instance variables will be created at the time of object creation and destroyed at the time of object destruction hence the scope of instance variables is exactly the same as scope of objects.
- Instance variables will be stored on the heap as the part of the object.
- Instance variables should be declared within the class directly but outside of any method or block or constructor.

Local variables

- Local variables will be stored inside the stack.
- The local variables will be created as part of the block execution in which it is declared and destroyed once that block execution completes. Hence the scope of the local variables is exactly the same as the scope of the block in which we declared.

Methods with Memory Map(JVM area)



Method Overloading

Method overloading enables several methods to use the same name but have distinct signatures, where the signature might vary based on the quantity, nature, or combination of input arguments.

- In the 'C' language we can't take 2 methods with the same name and different types. If there is a change in argument type compulsory we should go for a new method name.
  Example :
  abs() for int datatype
  labs() for long datatype
  fabs() for float datatype

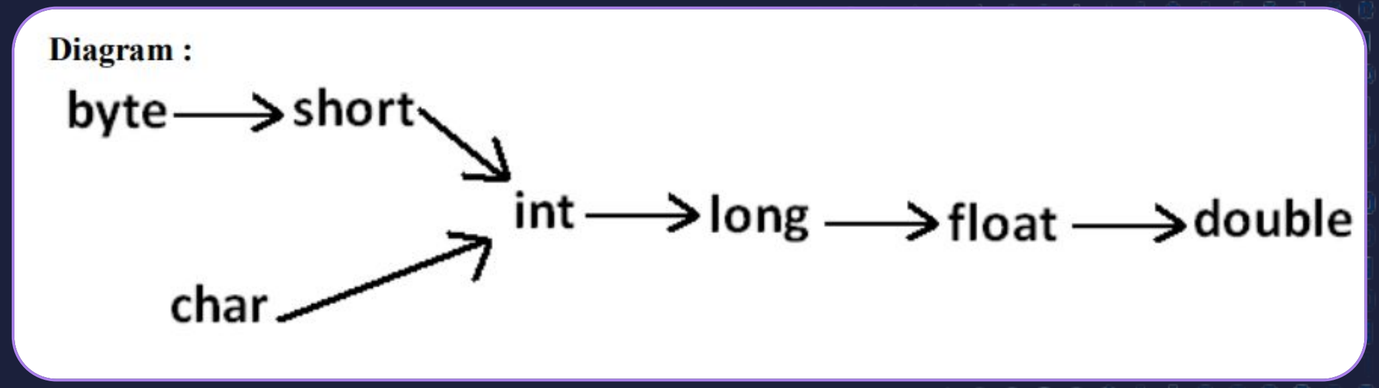
- Lack of overloading in "C" increases complexity of the programming.

- But in java we can take multiple methods with the same name and different argument types.
  abs(int) for int datatype
  abs(long) for long datatype
  abs(float) for float datatype

Automatic promotion in overloading

- In overloading if the compiler is unable to find the method with exact match we won't get any compile time error immediately.

The following are various possible automatic promotions in overloading.



Next Lecture

- Arrays in Java

THANK YOU