Language Fundamentals

By Rahul Barve

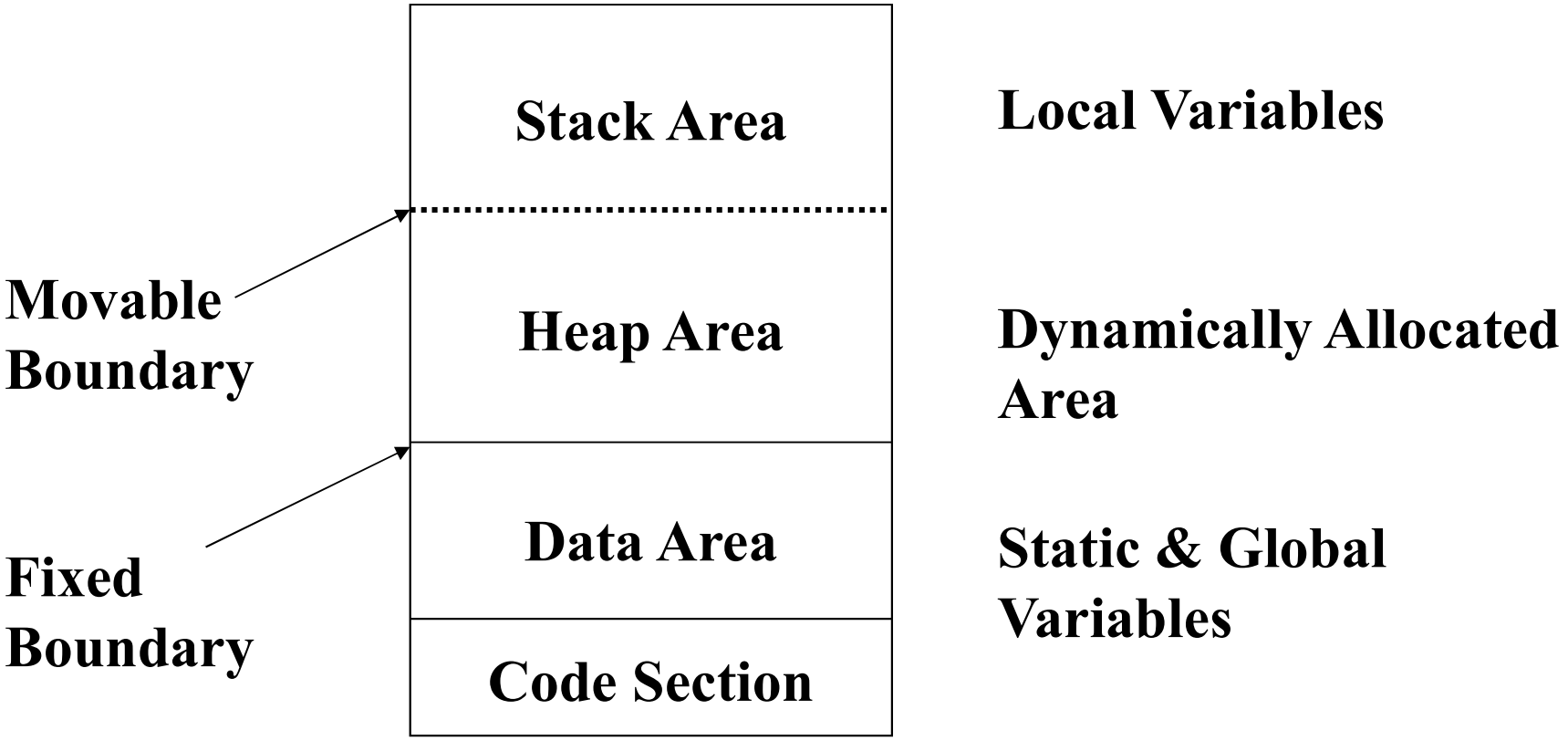
Objectives

- Memory Mapping
- Parameter Passing
- Working with Arrays.
- Garbage Collection

Memory Mapping

By Rahul Barve

Memory Mapping



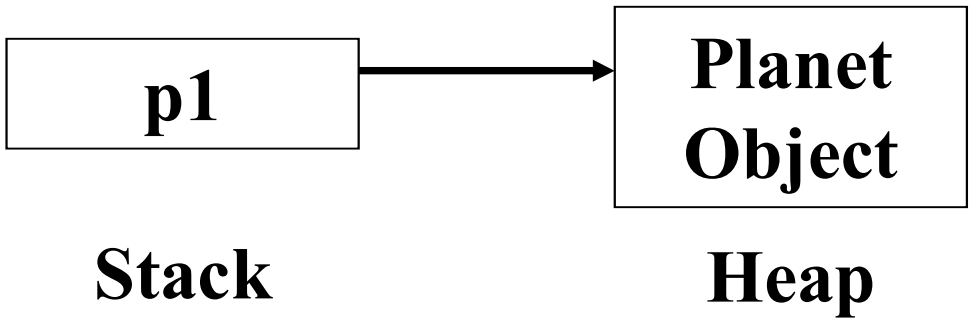
Memory Mapping

- In Java, object references are created on stack whereas the actual objects are created on heap.
- Java provides a `'new'` operator that allocates memory dynamically from heap area and returns the reference of the correct type.

Memory Mapping

- E.g.

```
Planet p1;
p1 = new Planet();
```

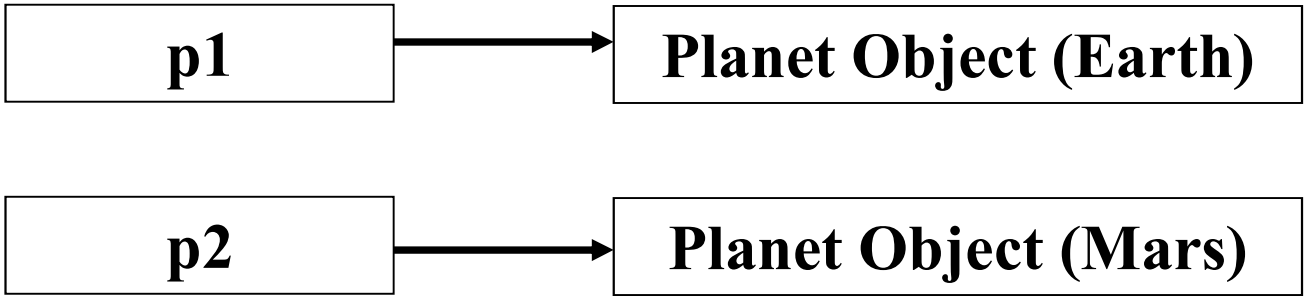


Memory Mapping

- An object may have multiple references but a reference can refer to only one object at a time.

Memory Mapping

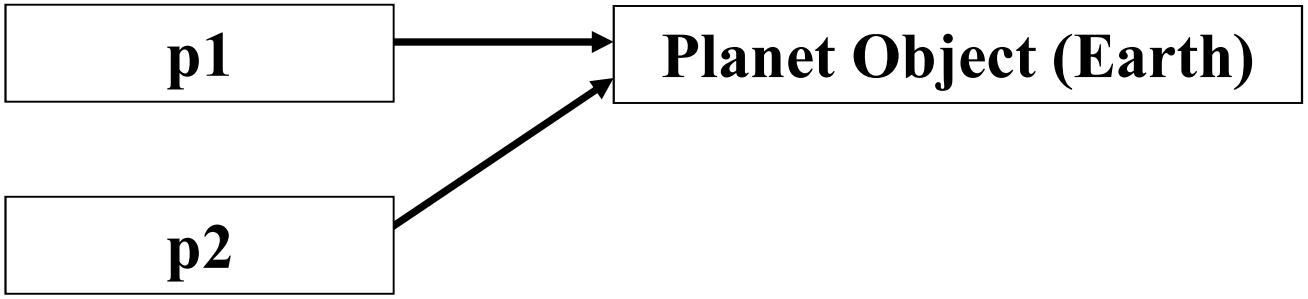
```
Planet p1 = new Planet("Earth");
Planet p2 = new Planet("Mars");
```



Stack **Heap**

Memory Mapping

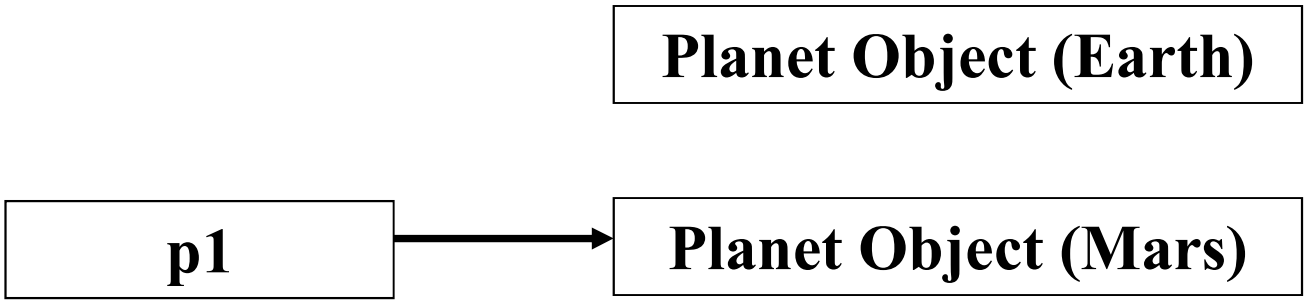
```
Planet p1 = new Planet("Earth");
Planet p2 = p1;
```



Stack **Heap**

Memory Mapping

```
Planet p1 = new Planet("Earth");
p1 = new Planet("Mars");
```



Parameter Passing

By Rahul Barve

Parameter Passing

- Methods can accept or return parameters either in the form of primitives or object types.
- While passing primitives, a copy of a variable is created on stack and hence primitives are always passed by value.

Parameter Passing

- While passing object types, a copy of a reference and not the actual object, is created on stack and hence objects are always passed by reference.

Arrays

By Rahul Barve

Arrays

- Array is a collection of similar typed elements, stored at contiguous memory locations.
- Array has a fixed dimension and the indexing starts from 0.

Arrays

- Arrays can be declared by 2 ways:
  - `int arr[] = new int[5];`
  - `int arr[] = {34,65,12};`

Arrays

- Java also provides support for dynamic arrays.
- E.g.

```
int size = 5;
int arr[] = new int[size];
```

Arrays

- Every array in Java is treated as an object.
- All array-type objects have a common property called as `length`.

Arrays

```
String skills[] =
      {“C”, “C++”, “Java”, “SQL”, “Pyhton”};
int size = skills.length;

for(int s=0;s<size;s++){
      String skill = skills[s]);
      System.out.println(skill);
}

OR

for(String skill : skills){
      System.out.println(skill);
}
```

Arrays

- It is also possible to create an array of object types.
- E.g.

```
Planet planets[] = new Planet[2];
planets[0] = new Planet("Earth");
planets[1] = new Planet("Mars");
```

Arrays

```
Planet p1 = new Planet("Earth");
Planet p2 = new Planet("Mars");
Planet planets[] = {p1, p2};
```