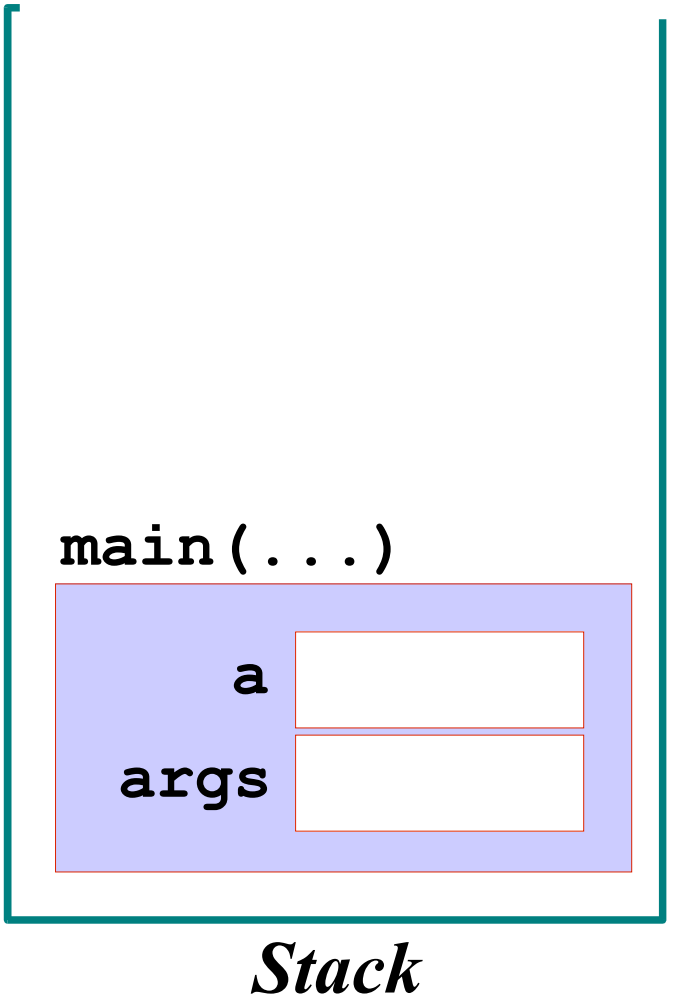
Example 1

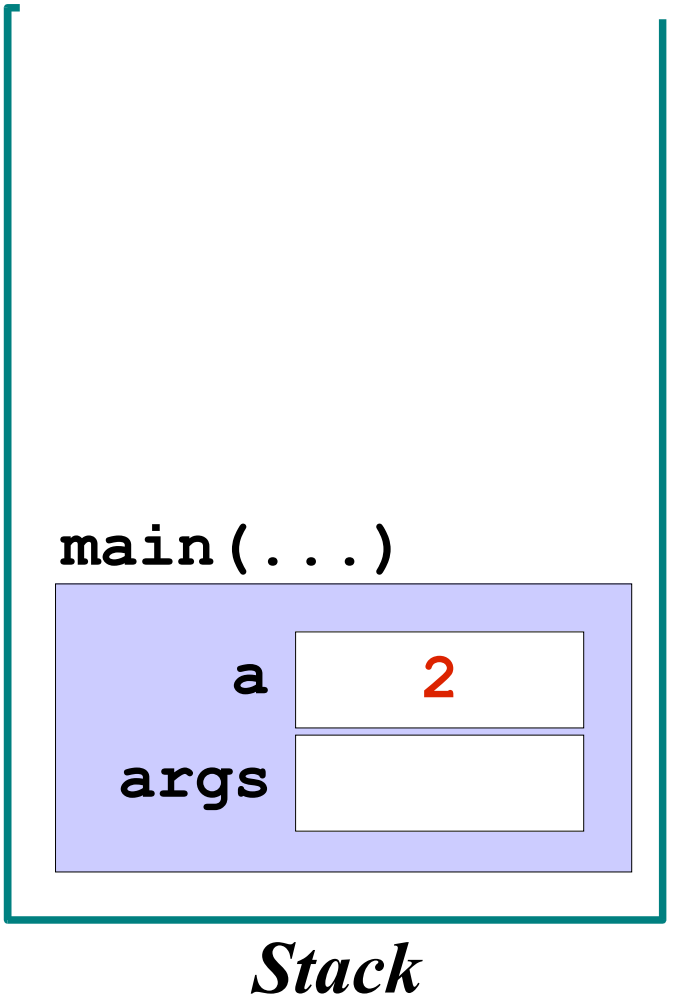
Managing Memory on Stack

```
public class Test1 {
   public static void f(int x) {
      int y = 1;
      y = x + y;
      int z = 10;
      x = y * z;
   }

   public static void main(String[] args) {
      int a = 2;
      f(a);
      System.out.println(a);
   }
}
```

```
public class Test1 {
   public static void f(int x) {
      int y = 1;
      y = x + y;
      int z = 10;
      x = y * z;
   }

   public static void main(...) {
      int a = 2;
      f(a);
      System.out.println(a);
   }
}
```

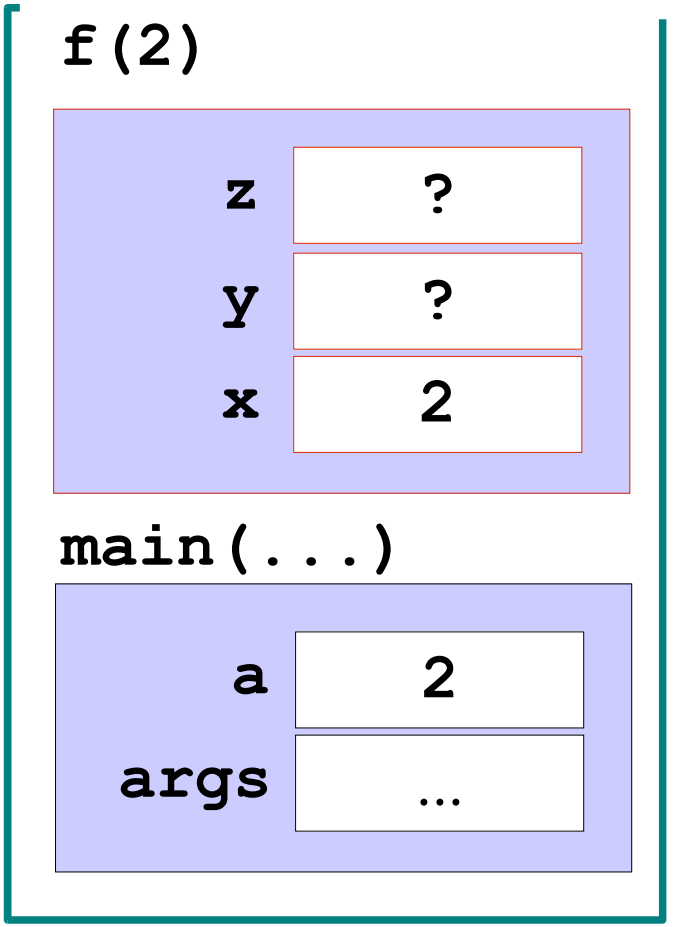


```
public class Test1 {
   public static void f(int x) {
      int y = 1;
      y = x + y;
      int z = 10;
      x = y * z;
   }

   public static void main(...) {
      int a = 2;
      f(a);
      System.out.println(a);
   }
}
```



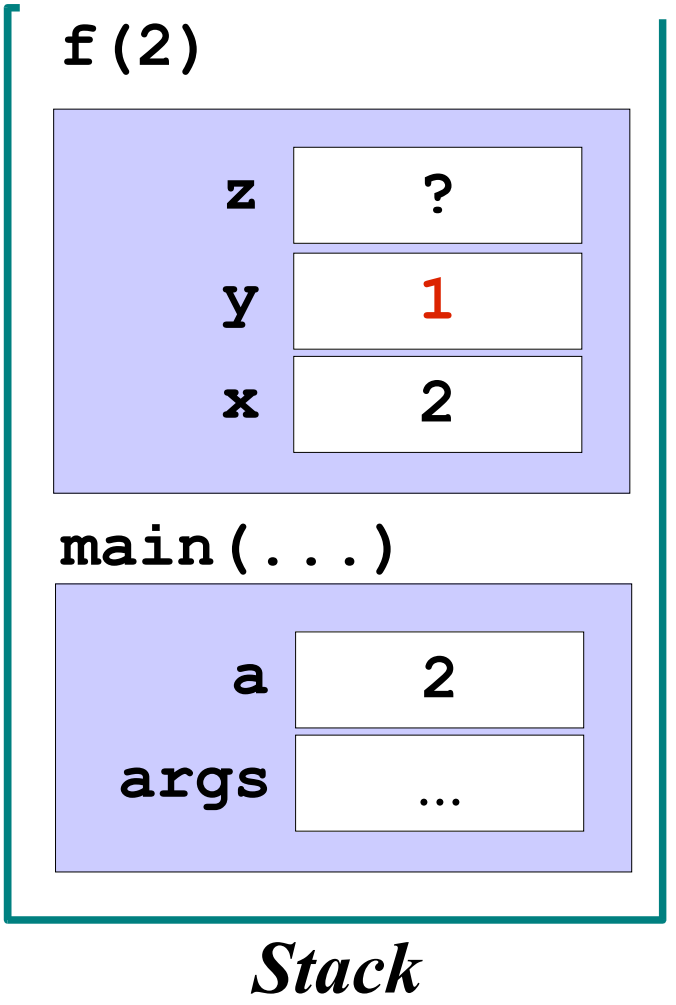
Stack

```
public class Test1 {
   public static void f(int x) {
      int y = 1;
      y = x + y;
      int z = 10;
      x = y * z;
   }

   public static void main(...) {
      int a = 2;
      f(a);
      System.out.println(a);
   }
}
```



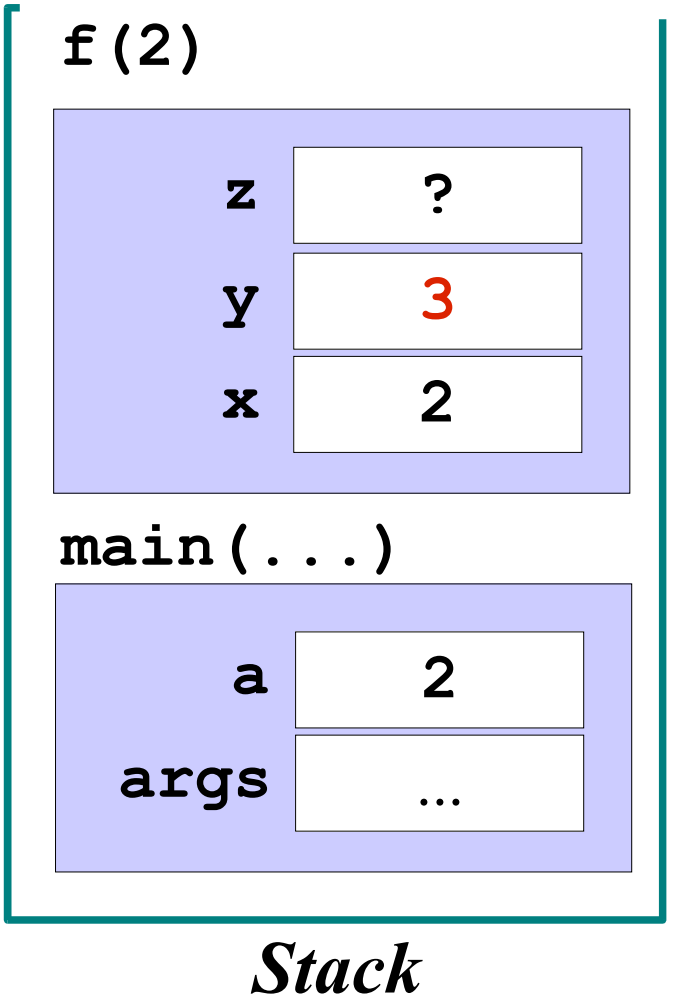
Stack

```
public class Test1 {
   public static void f(int x) {
      int y = 1;
      y = x + y;
      int z = 10;
      x = y * z;
   }

   public static void main(...) {
      int a = 2;
      f(a);
      System.out.println(a);
   }
}
```

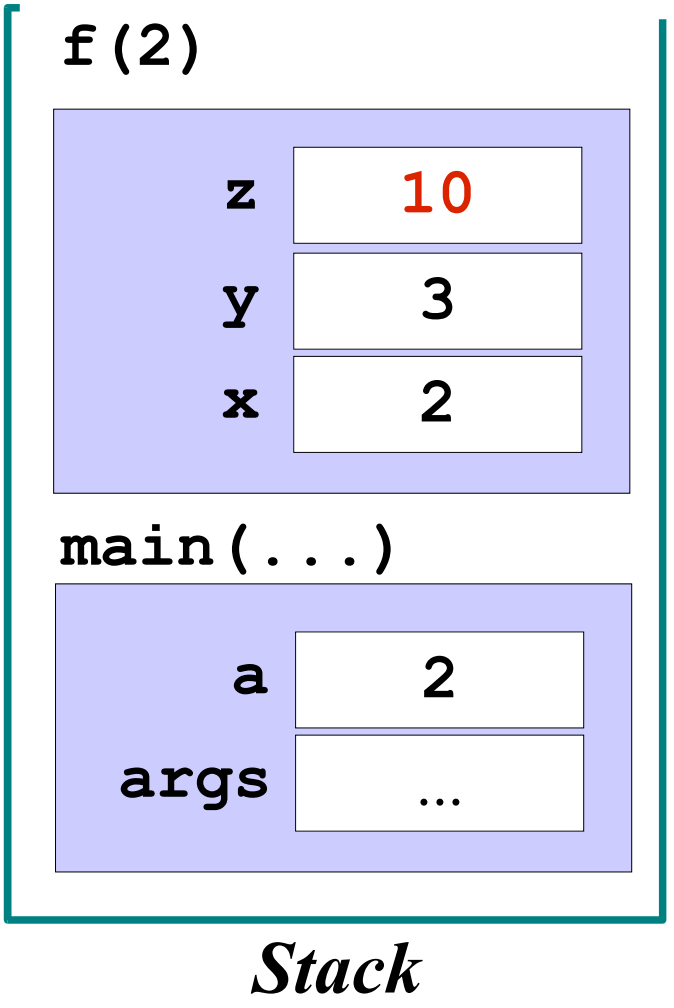


Stack

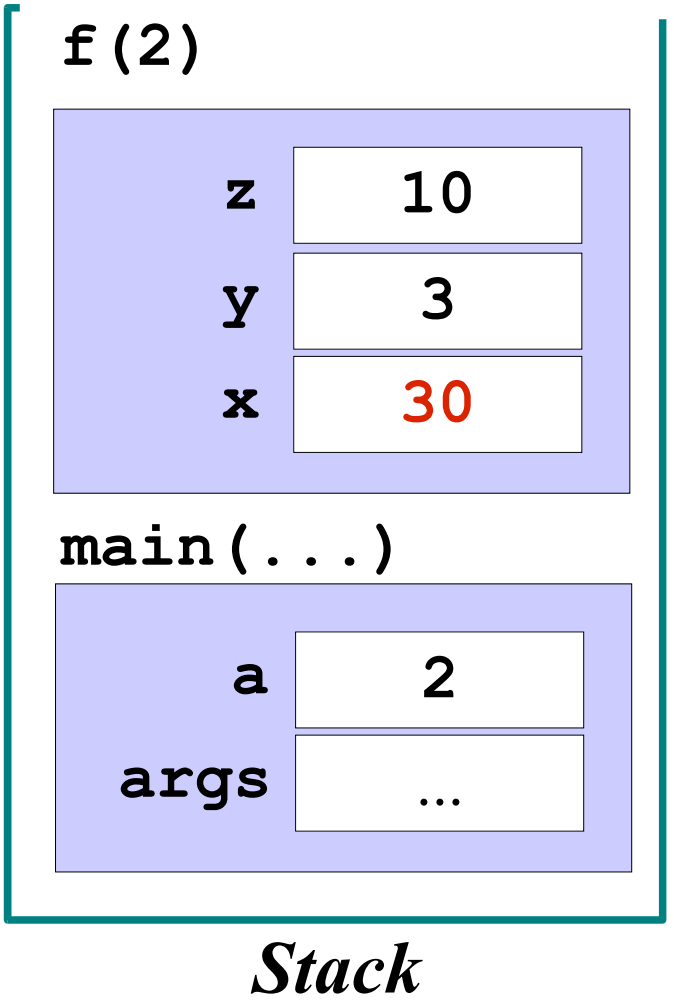
```
public class Test1 {
   public static void f(int x) {
      int y = 1;
      y = x + y;
      int z = 10;
      x = y * z;
   }

   public static void main(...) {
      int a = 2;
      f(a);
      System.out.println(a);
   }
}
```



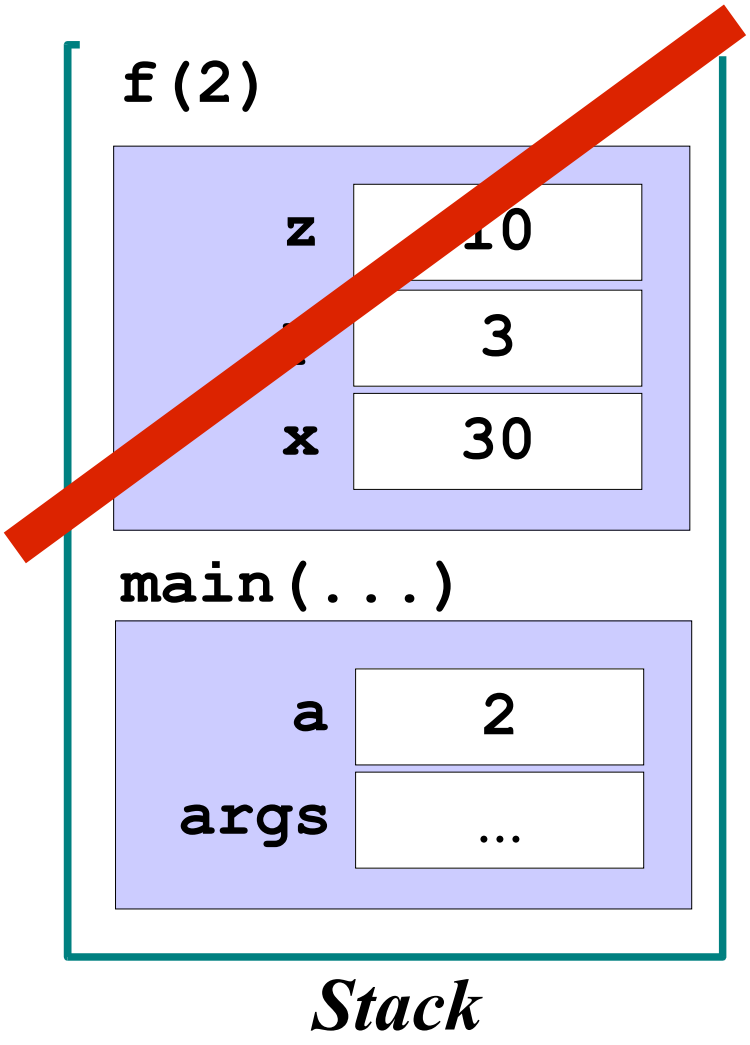
```
public class Test1 {
   public static void f(int x) {
      int y = 1;
      y = x + y;
      int z = 10;
      x = y * z;
   }

   public static void main(...) {
      int a = 2;
      f(a);
      System.out.println(a);
   }
}
```



```
public class Test1 {
   public static void f(int x) {
      int y = 1;
      y = x + y;
      int z = 10;
      x = y * z;
   }

   public static void main(...) {
      int a = 2;
      f(a);
      System.out.println(a);
   }
}
```

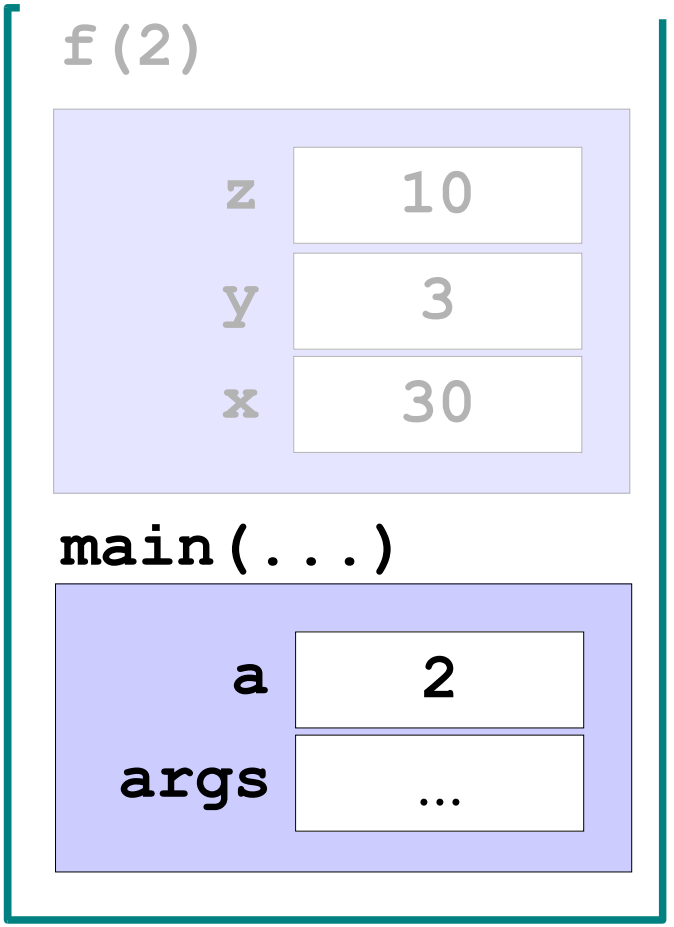


```
public class Test1 {
   public static void f(int x) {
      int y = 1;
      y = x + y;
      int z = 10;
      x = y * z;
   }

   public static void main(...) {
      int a = 2;
      f(a);
      System.out.println(a);
   }
}
```



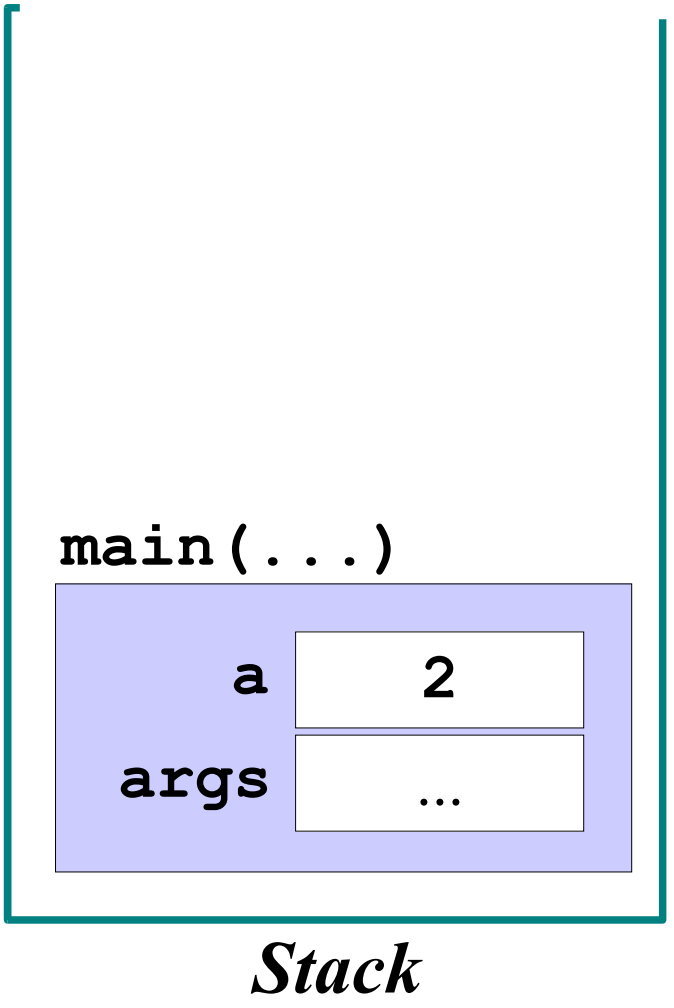
Stack

```
public class Test1 {
   public static void f(int x) {
      int y = 1;
      y = x + y;
      int z = 10;
      x = y * z;
   }

   public static void main(...) {
      int a = 2;
      f(a);
      System.out.println(a);
   }
}
```

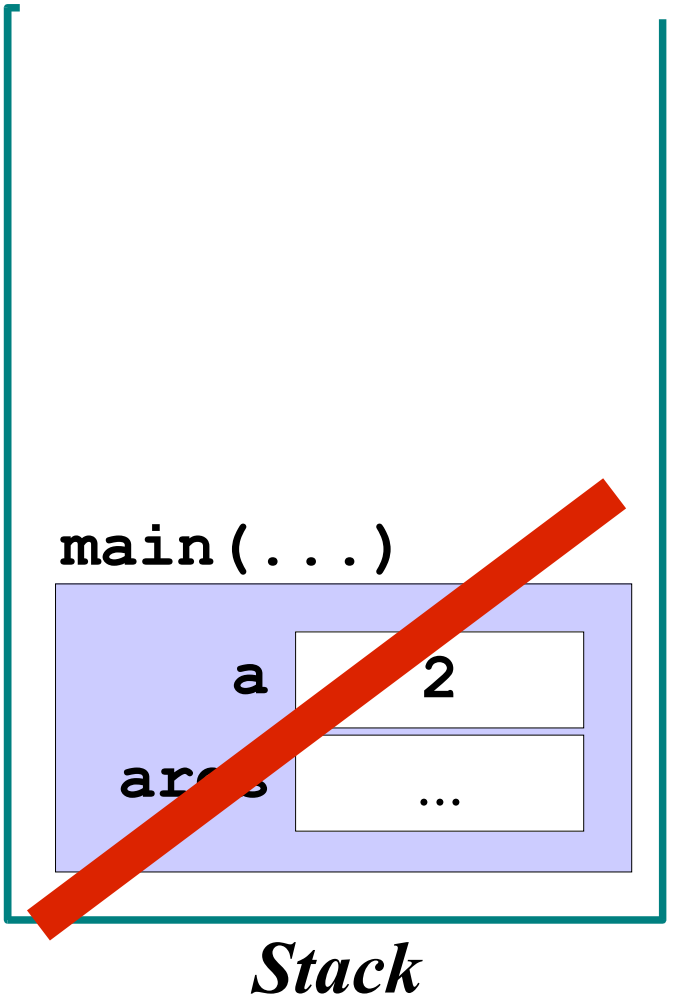


Stack

```
public class Test1 {
   public static void f(int x) {
      int y = 1;
      y = x + y;
      int z = 10;
      x = y * z;
   }

   public static void main(...) {
      int a = 2;
      f(a);
      System.out.println(a);
   }
}
```

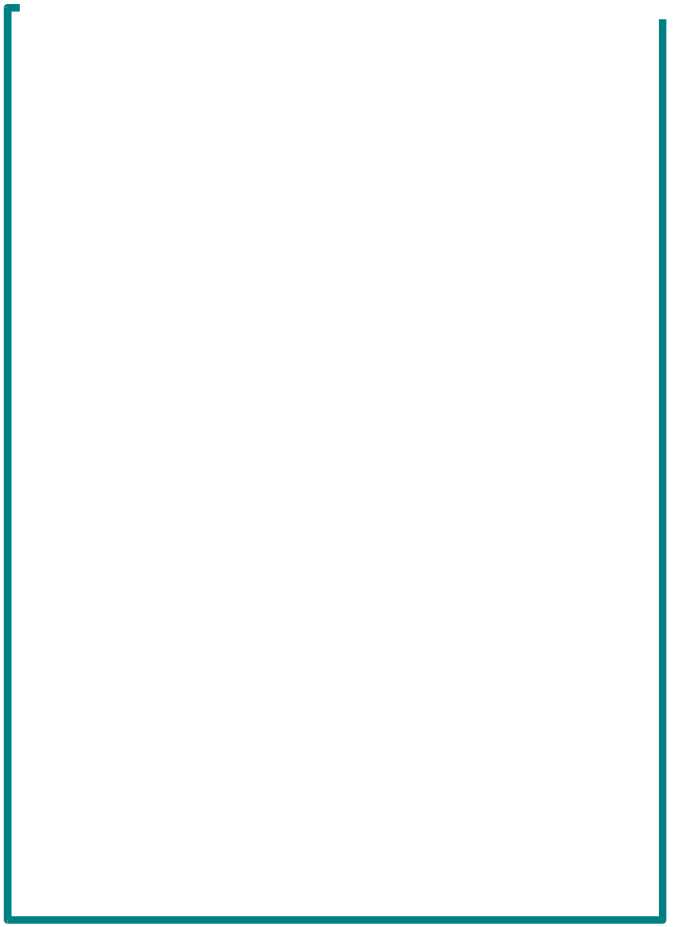


```
public class Test1 {
   public static void f(int x) {
      int y = 1;
      y = x + y;
      int z = 10;
      x = y * z;
   }

   public static void main(...) {
      int a = 2;
      f(a);
      System.out.println(a);
   }
}
```



Stack

```
public class Test1 {
   public static void f(int x) {
      int y = 1;
      y = x + y;
      int z = 10;
      x = y * z;
   }

   public static void main(...) {
      int a = 2;
      f(a);
      System.out.println(a);
   }
}
```

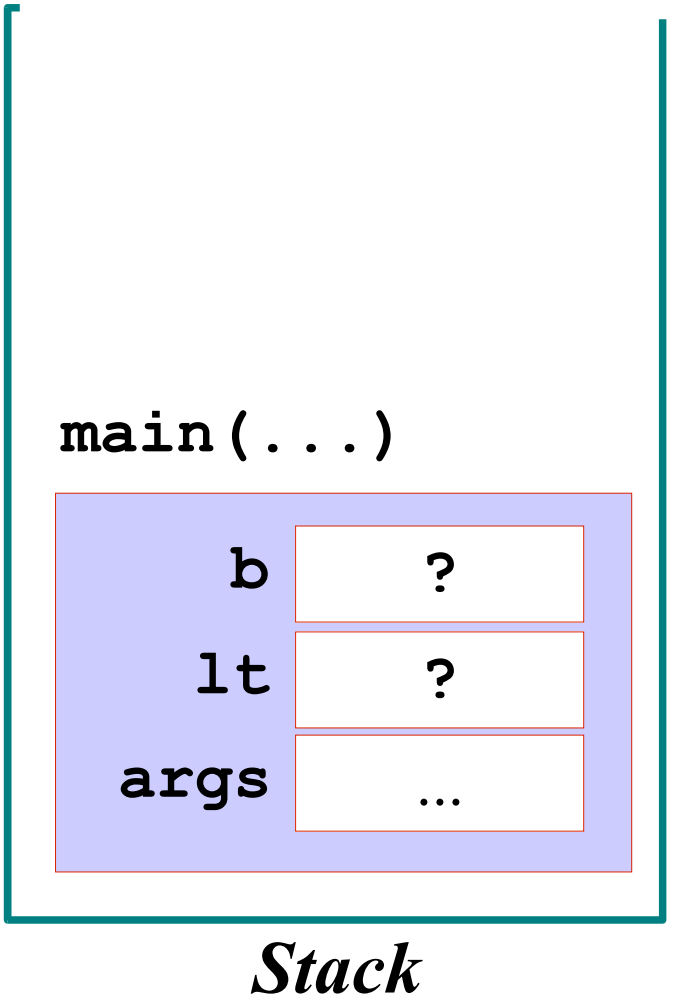


Stack

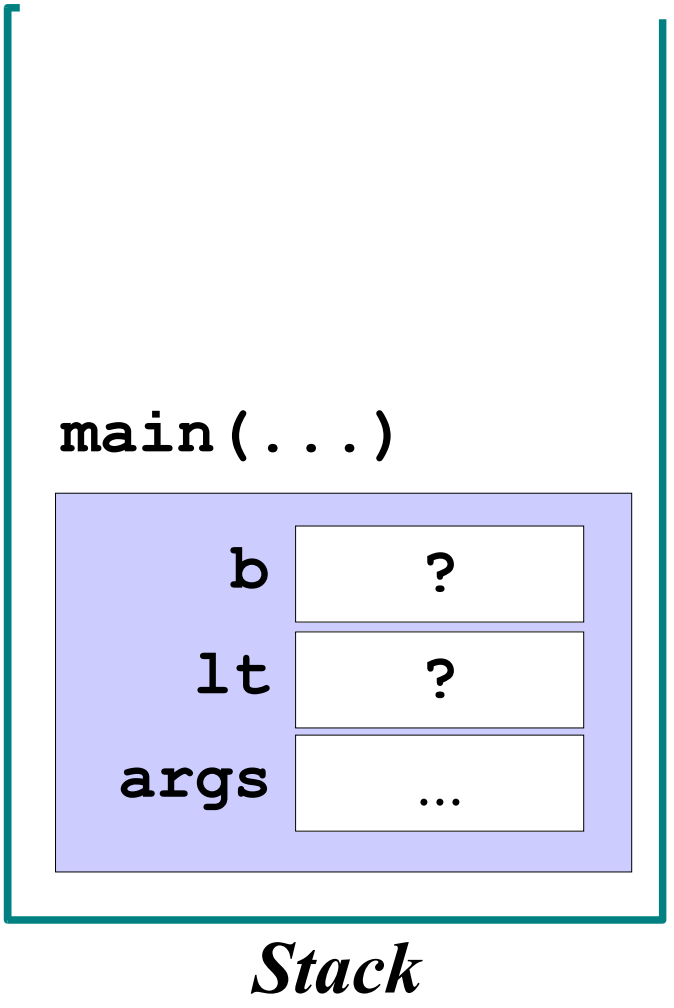
Example 2

Managing Memory in Heap

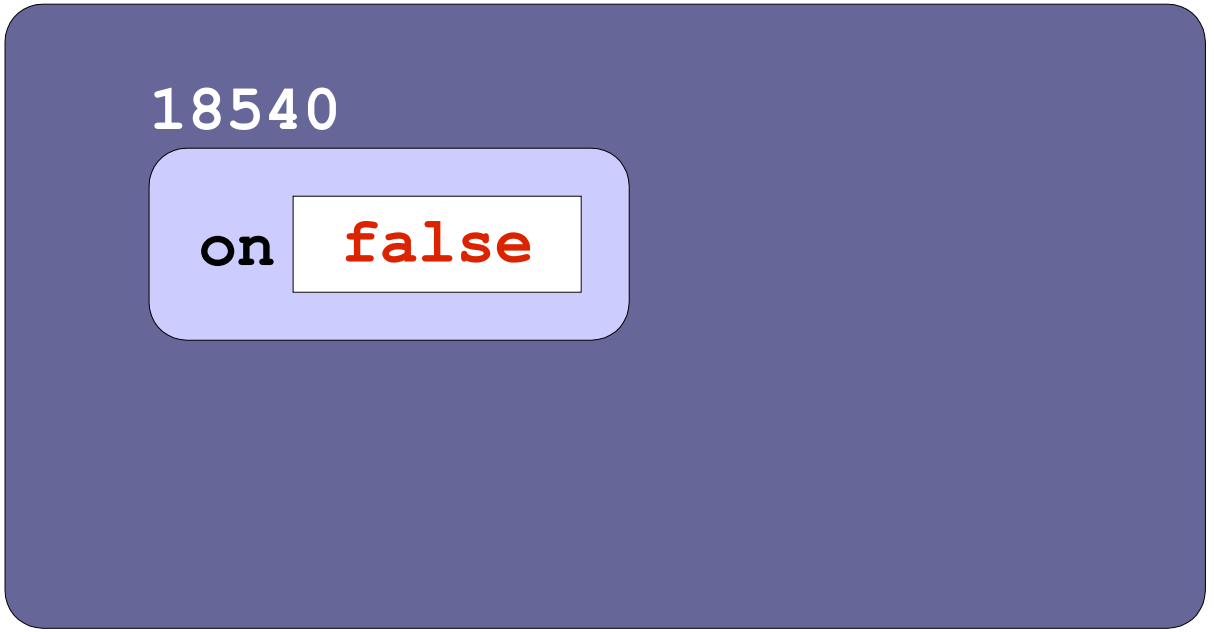
```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      boolean b;
      b = lt.isOn();
      System.out.println(b);
   }
}
```



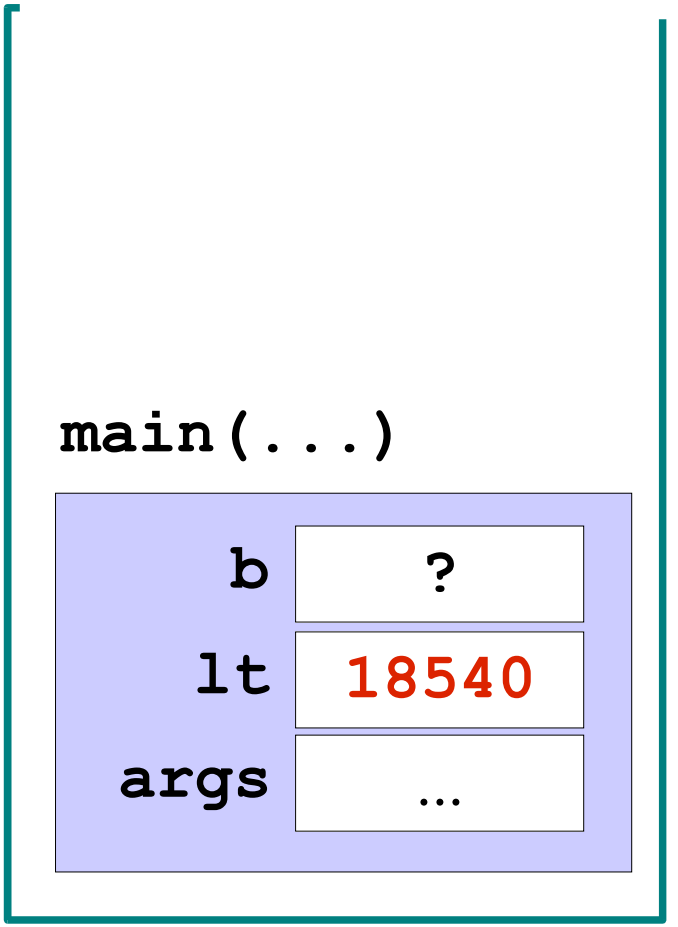
```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      boolean b;
      b = lt.isOn();
      System.out.println(b);
   }
}
```



```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      boolean b;
      b = lt.isOn();
      System.out.println(b);
   }
}
```

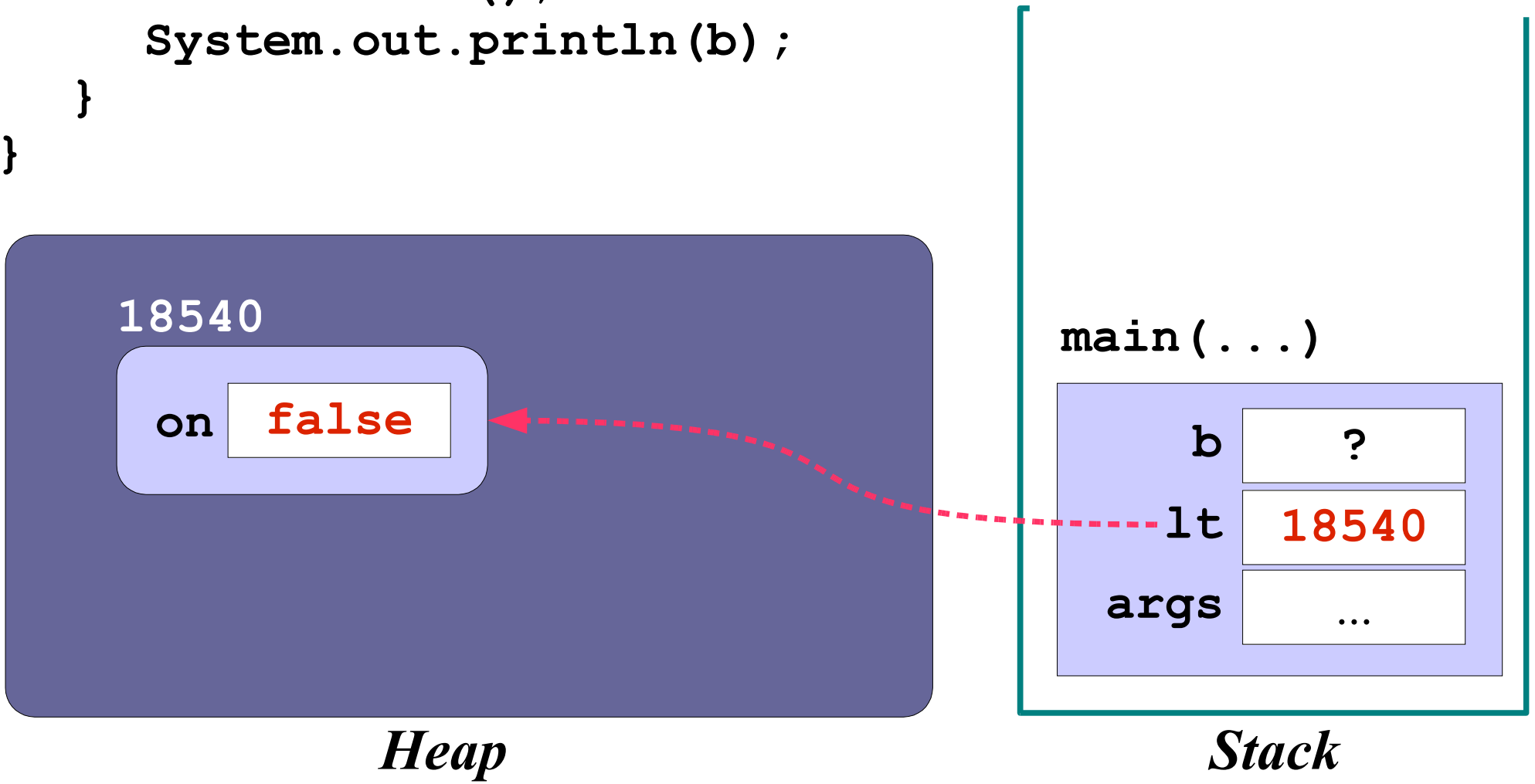


Heap

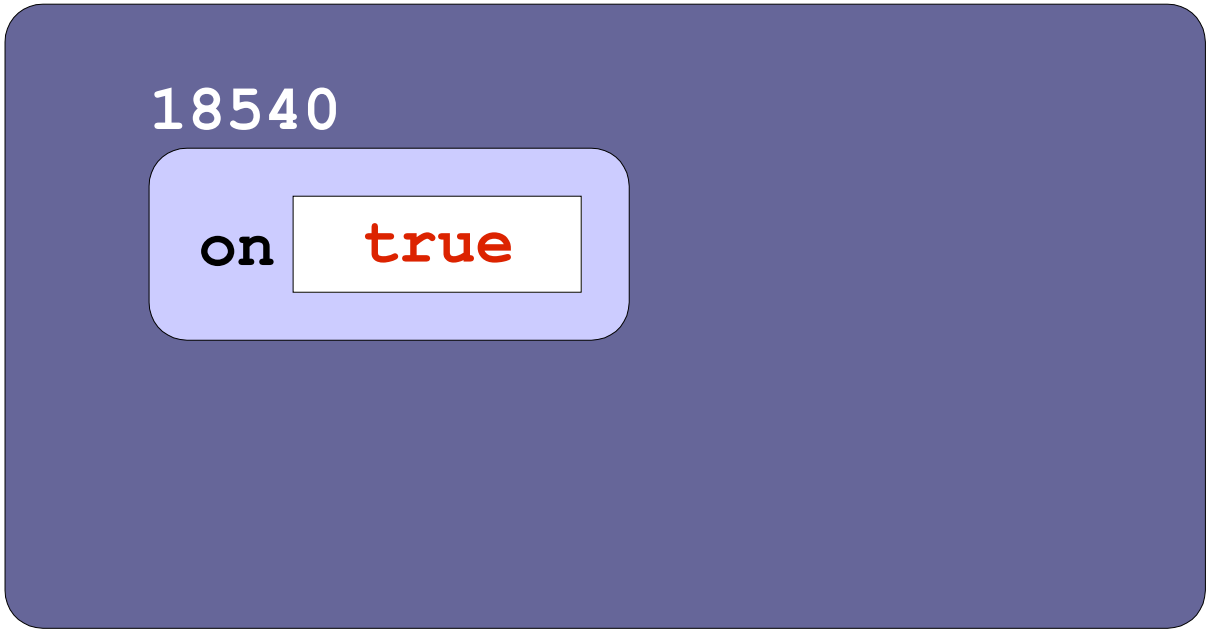


Stack

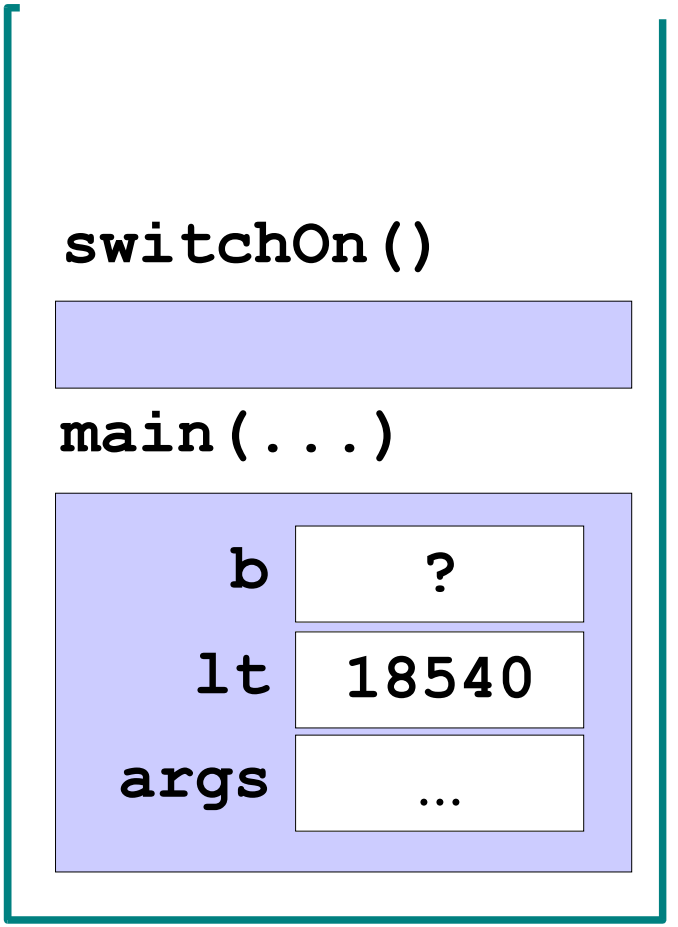
```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      boolean b;
      b = lt.isOn();
      System.out.println(b);
   }
}
```



```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      boolean b;
      b = lt.isOn();
      System.out.println(b);
   }
}
```

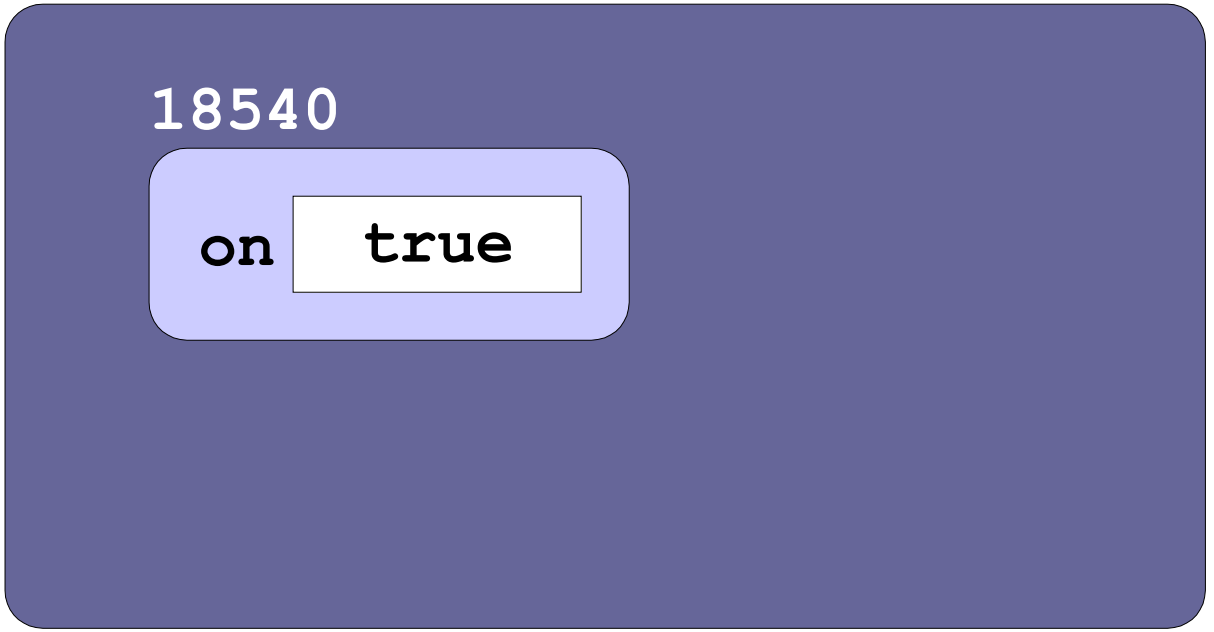


Heap

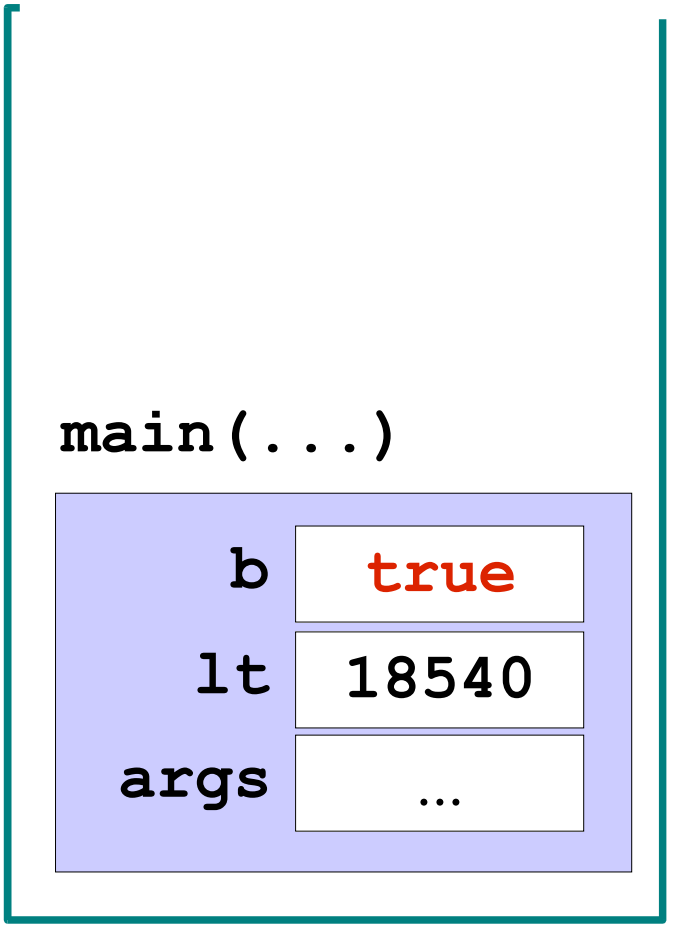


Stack

```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      boolean b;
      b = lt.isOn();
      System.out.println(b);
   }
}
```



Heap

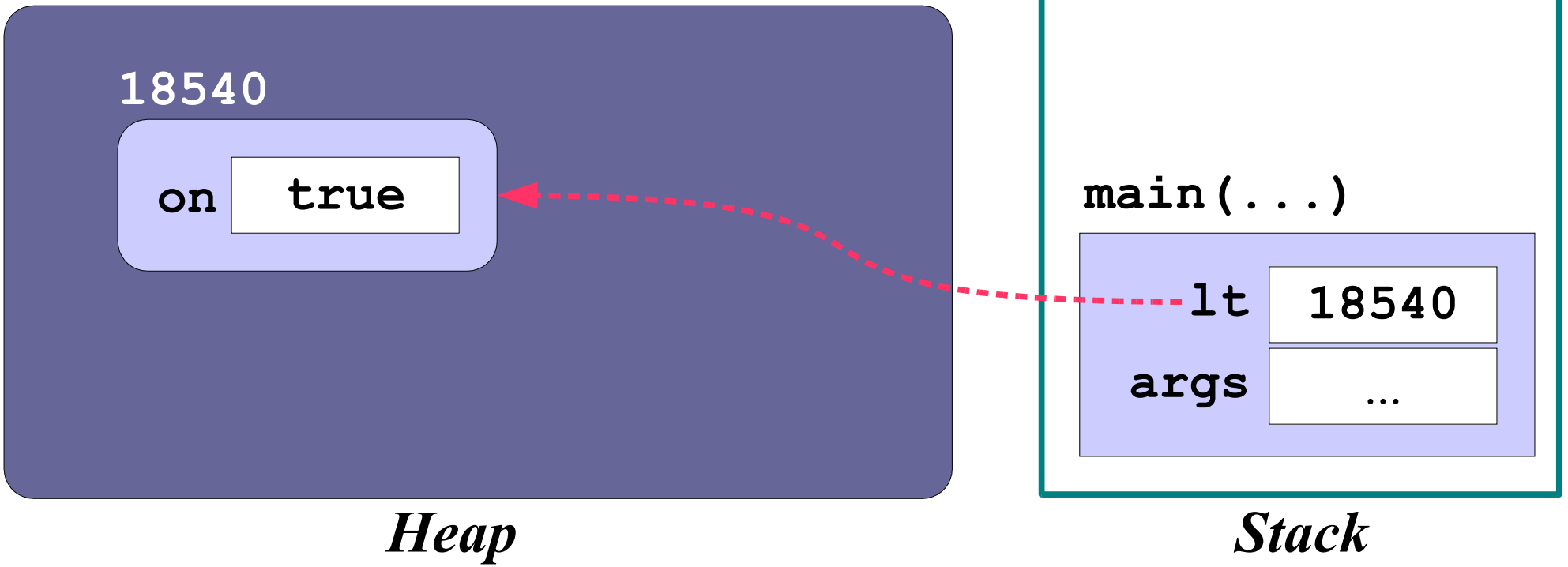


Stack

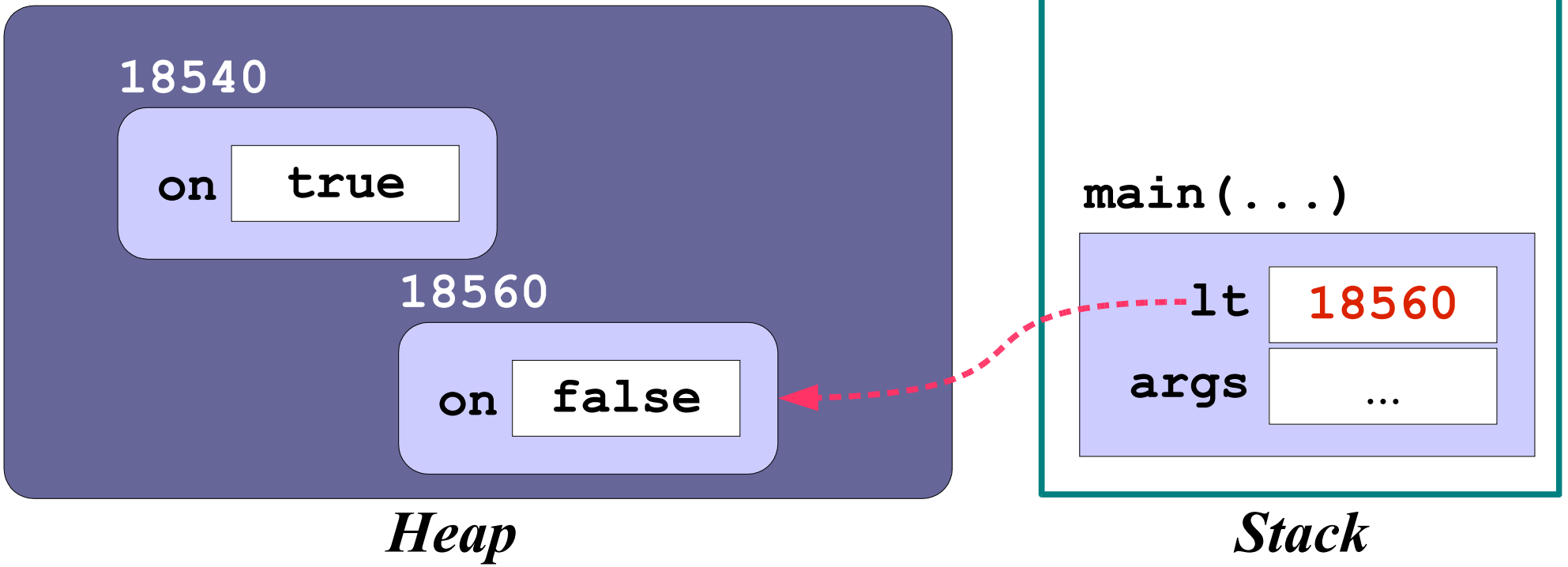
Example 3

Generating Garbage

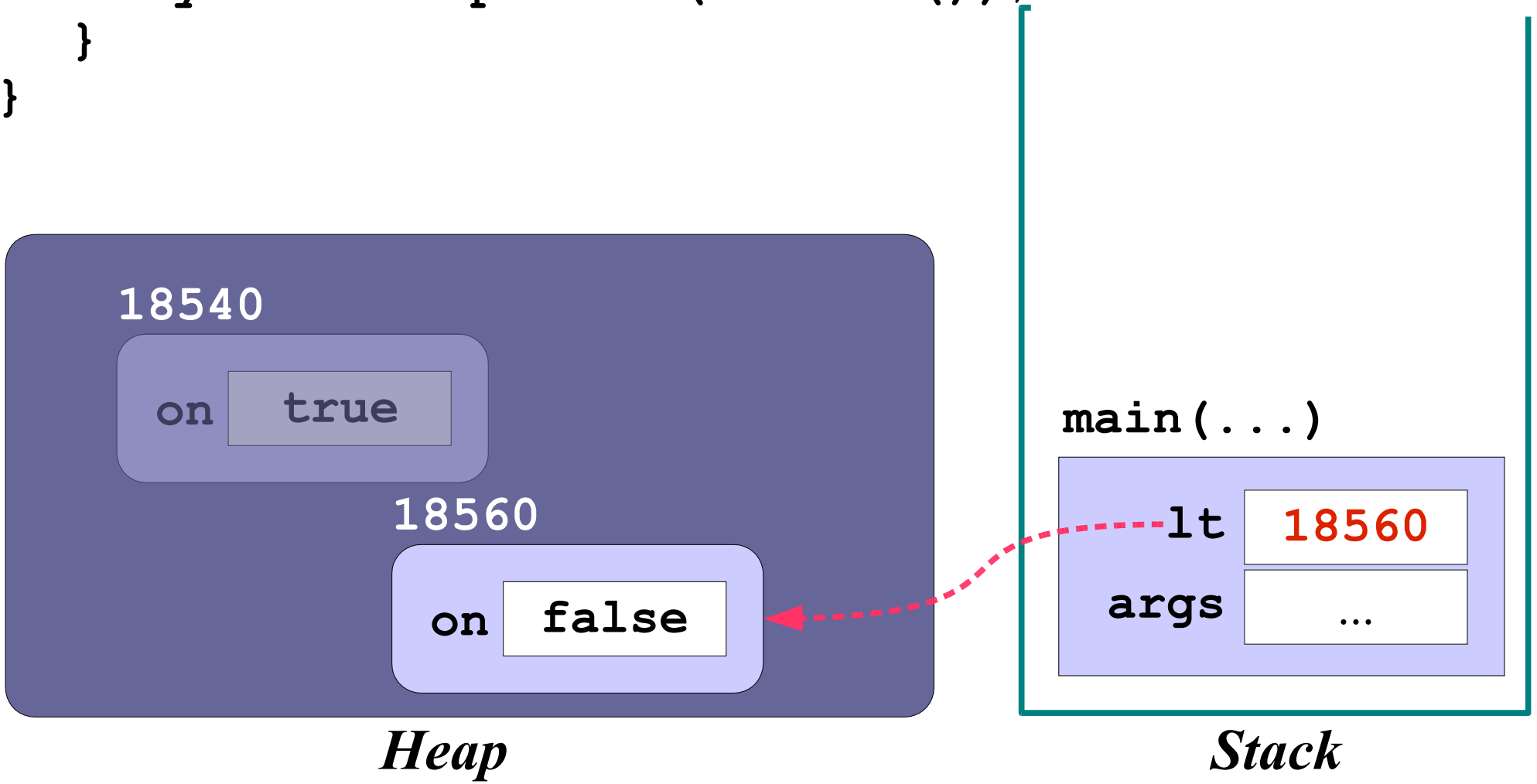
```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      lt = new Light();
      System.out.println(lt.isOn());
   }
}
```



```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      lt = new Light();
      System.out.println(lt.isOn());
   }
}
```



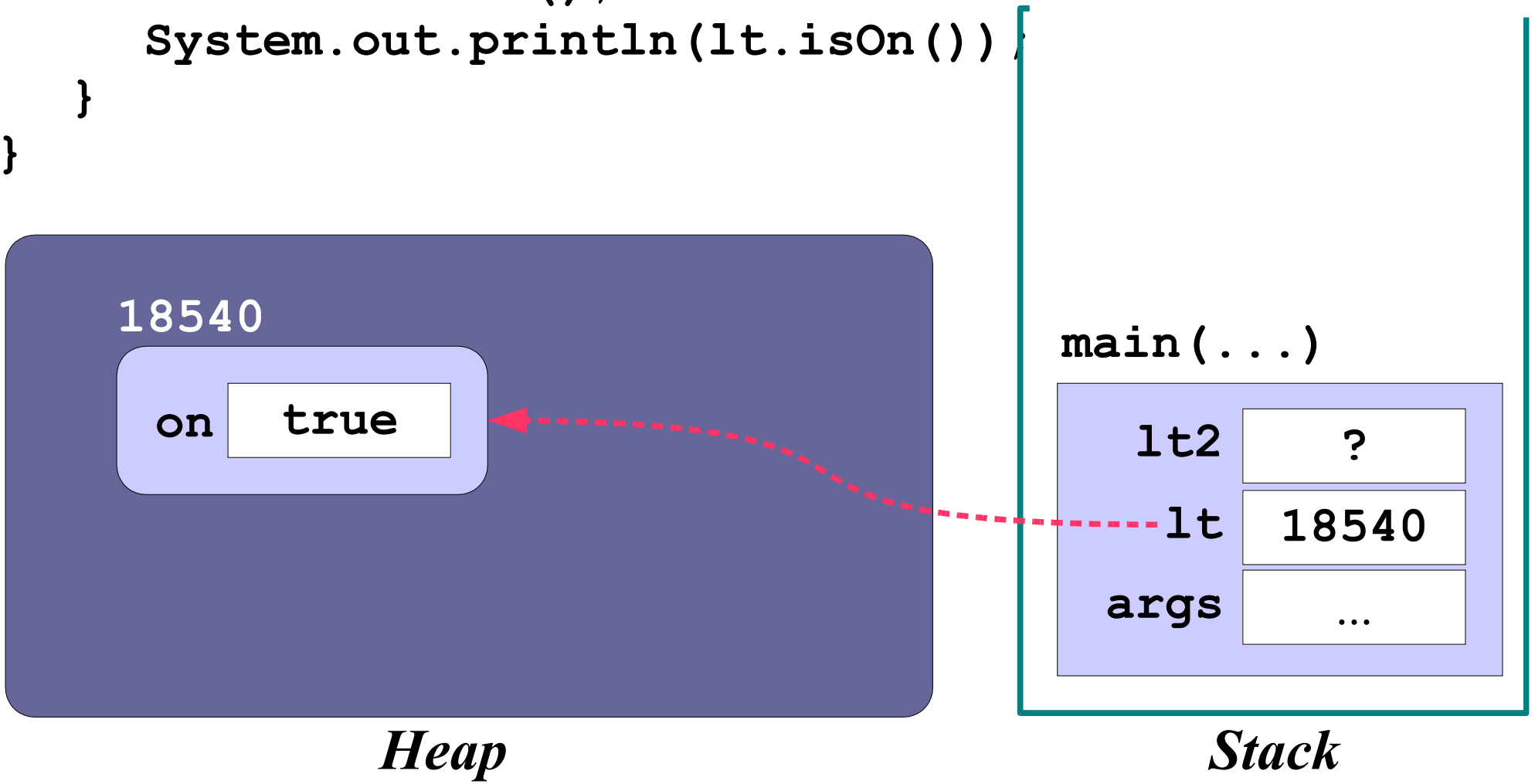
```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      lt = new Light();
      System.out.println(lt.isOn());
   }
}
```



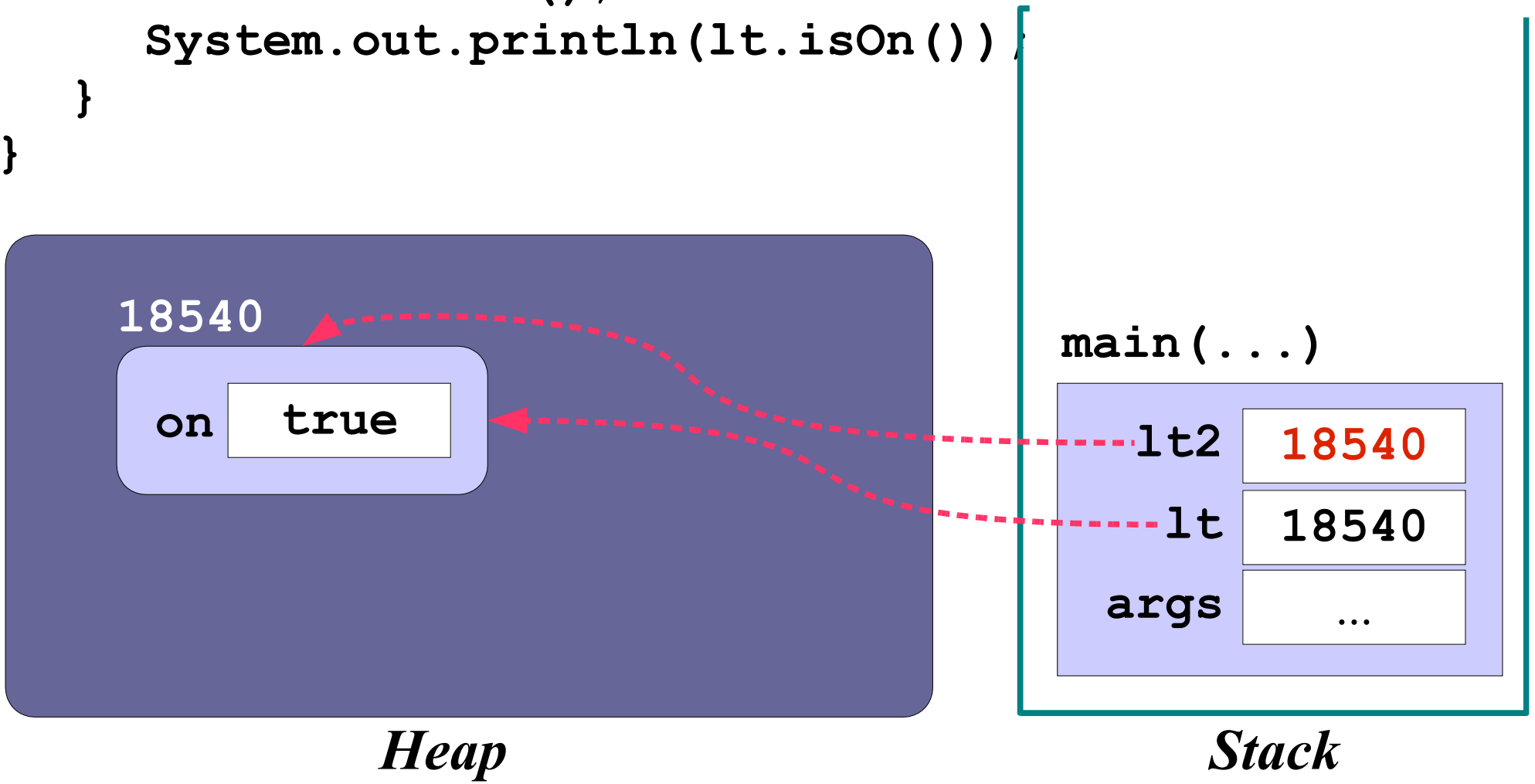
Example 4

Aliasing

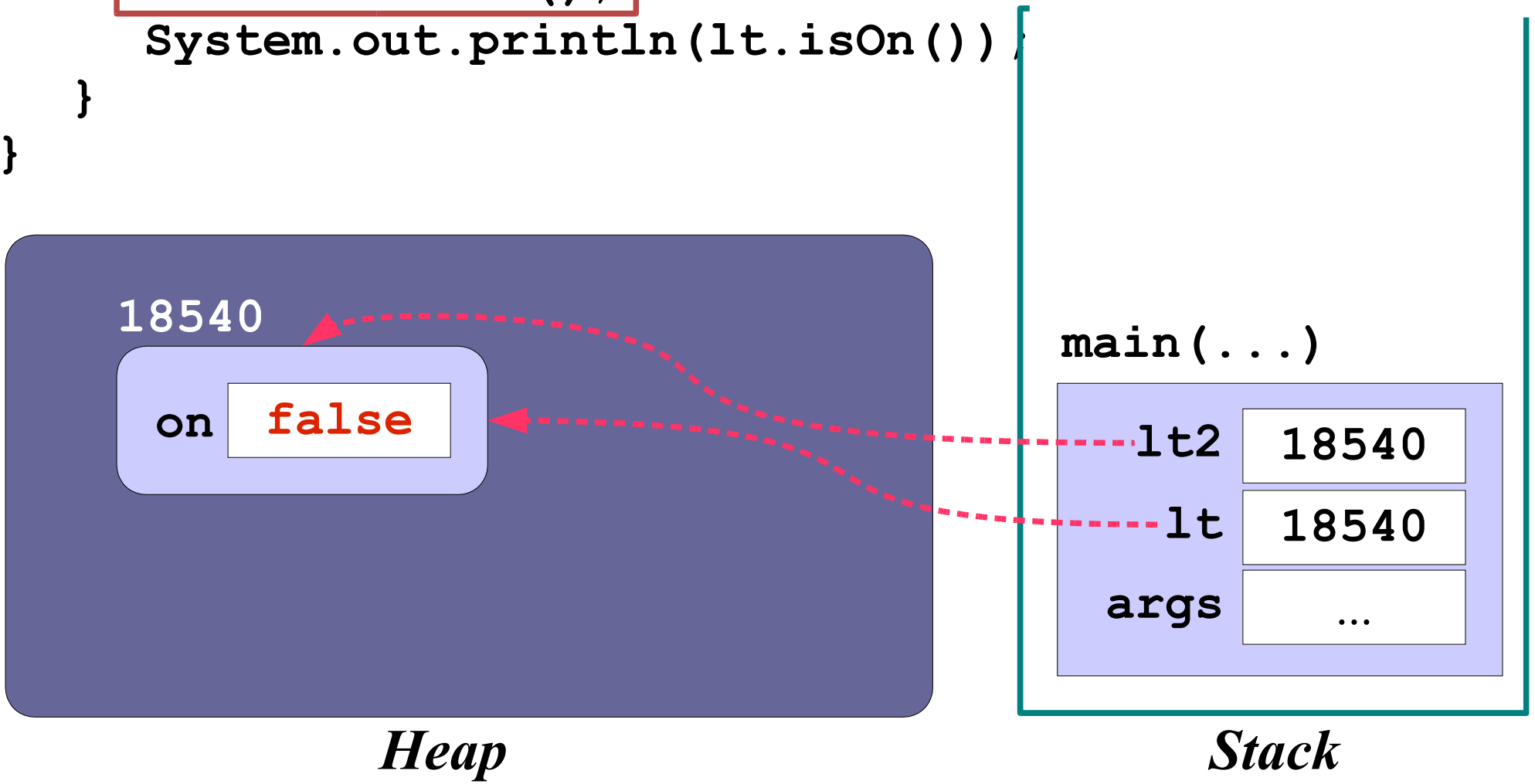
```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      Light lt2 = lt;
      lt2.switchOff();
      System.out.println(lt.isOn());
   }
}
```



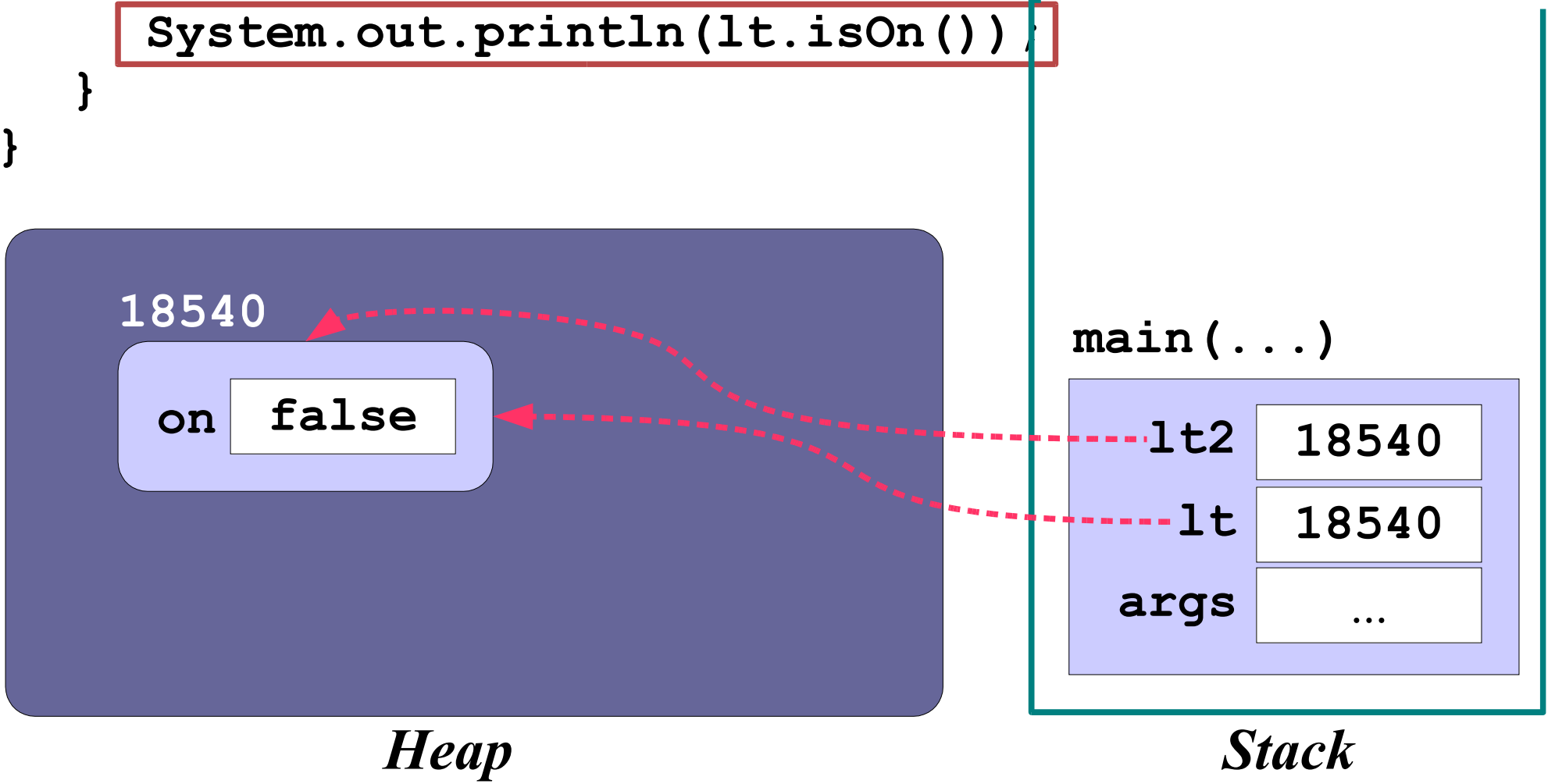
```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      Light lt2 = lt;
      lt2.switchOff();
      System.out.println(lt.isOn());
   }
}
```



```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      Light lt2 = lt;
      lt2.switchOff();
      System.out.println(lt.isOn());
   }
}
```



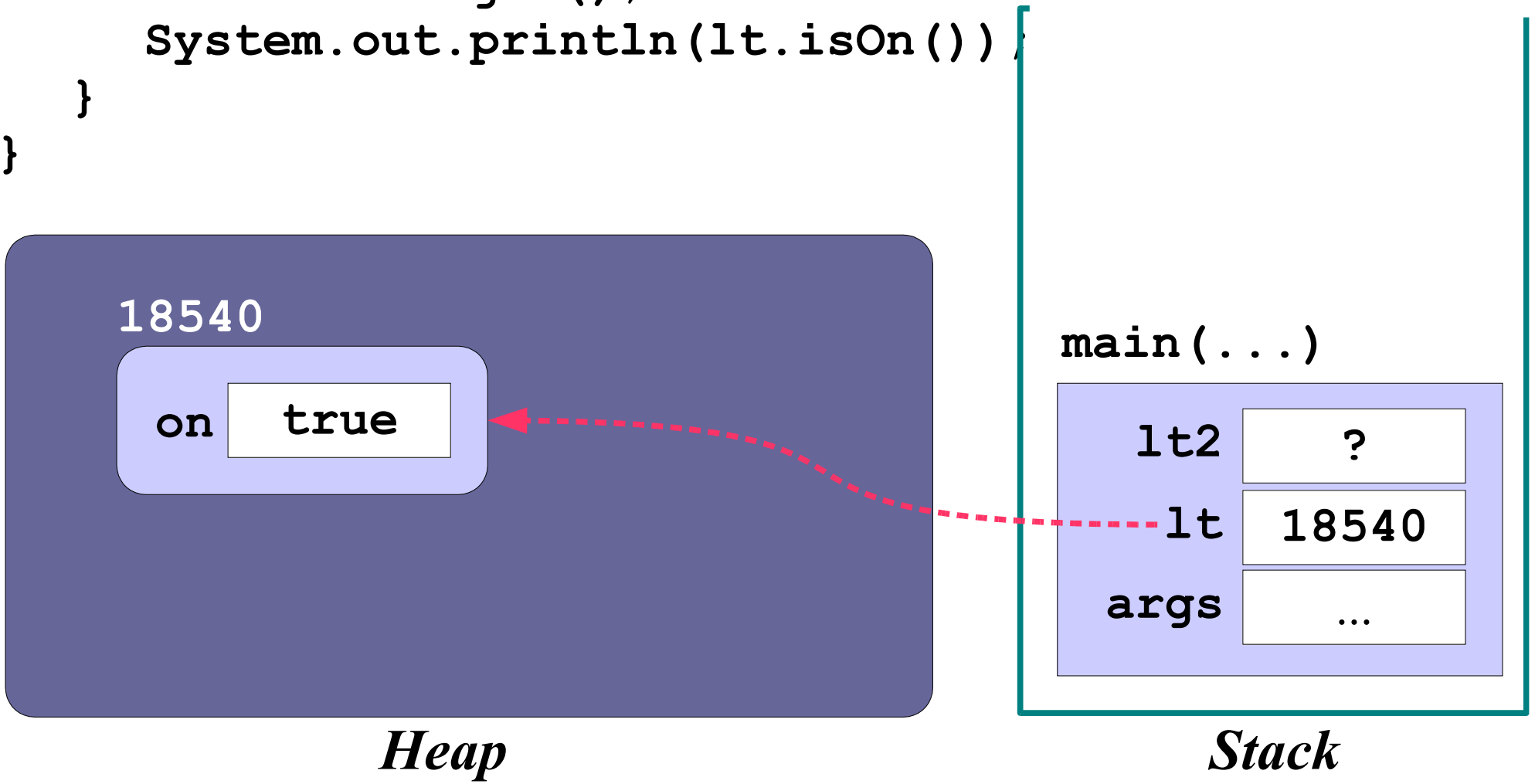
```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      Light lt2 = lt;
      lt2.switchOff();
      System.out.println(lt.isOn());
   }
}
```



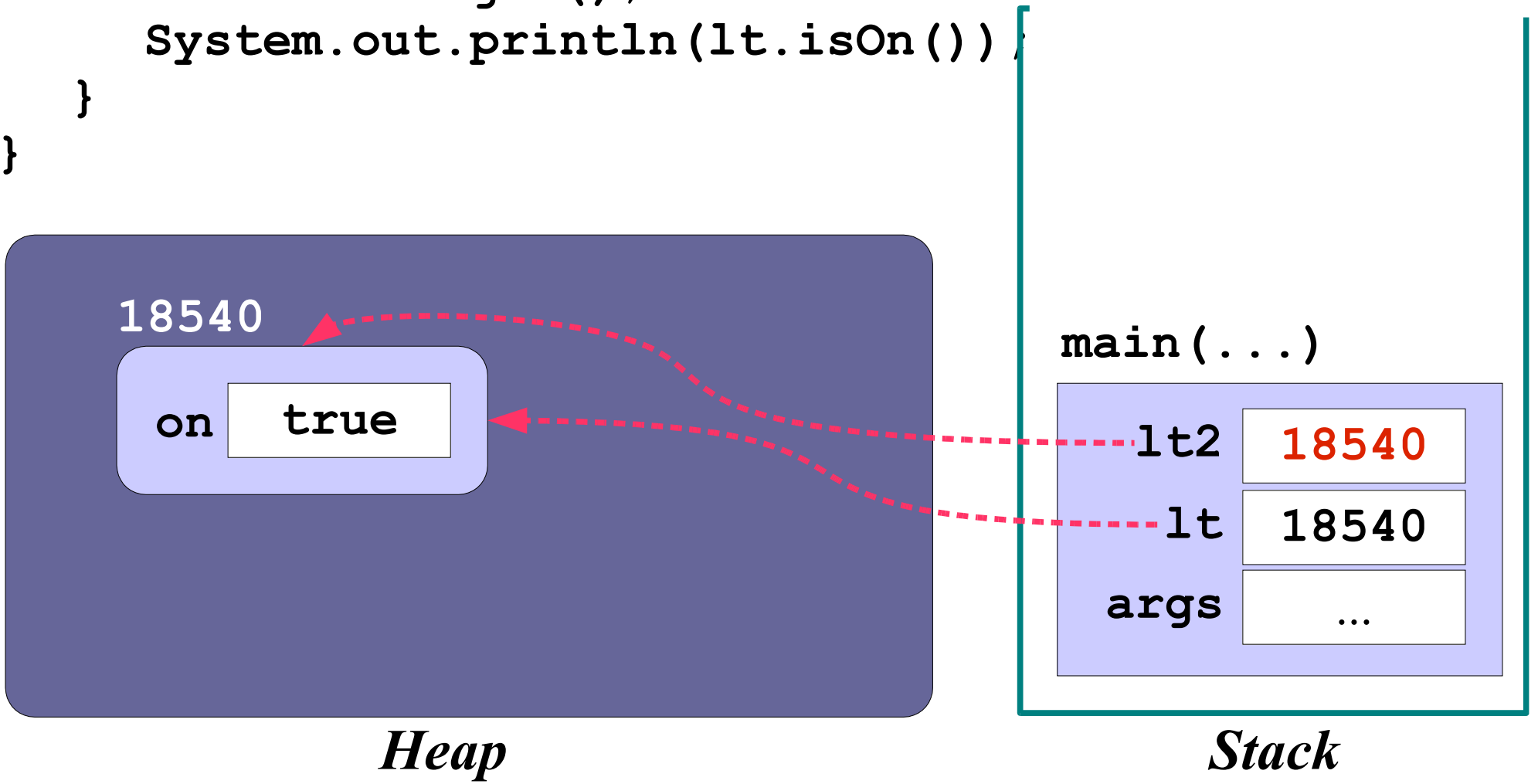
Example 5

Avoiding Garbage

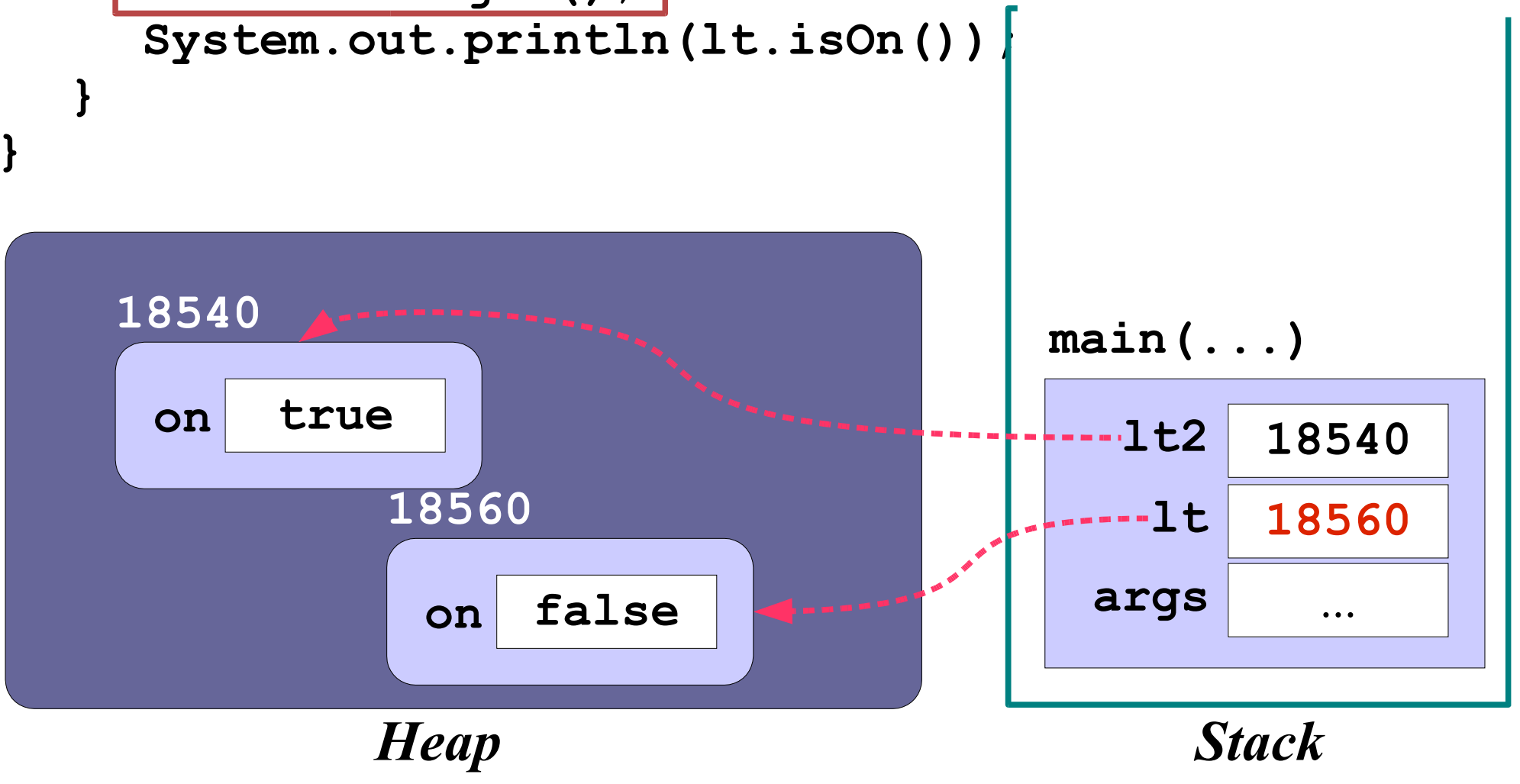
```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      Light lt2 = lt;
      lt = new Light();
      System.out.println(lt.isOn());
   }
}
```



```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      Light lt2 = lt;
      lt = new Light();
      System.out.println(lt.isOn());
   }
}
```



```
public class LightProg {
   public static void main(...) {
      Light lt;
      lt = new Light();
      lt.switchOn();
      Light lt2 = lt;
      lt = new Light();
      System.out.println(lt.isOn());
   }
}
```

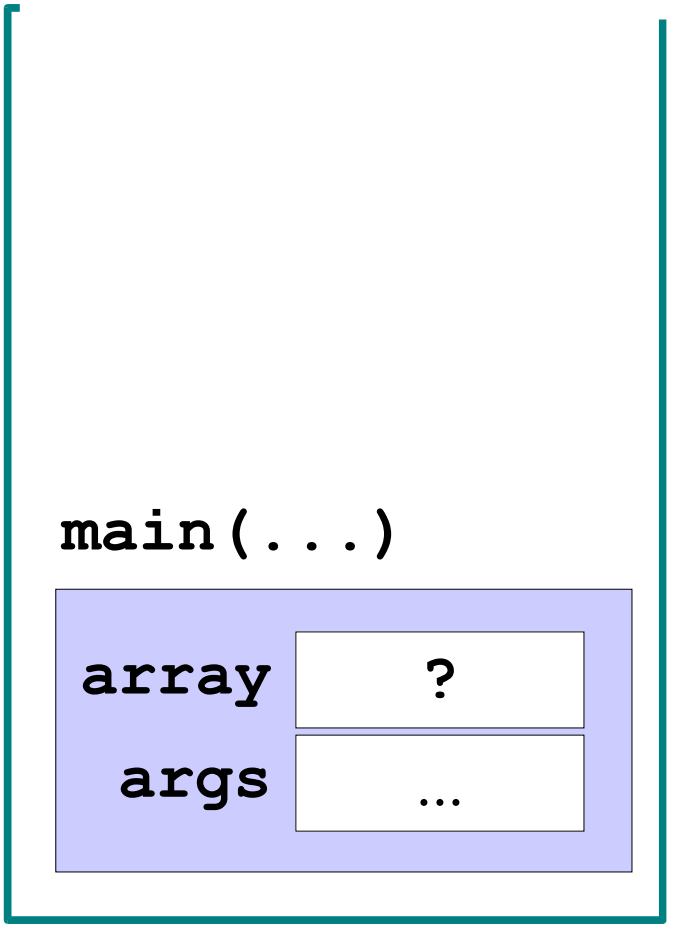


Example 6

Allocating Arrays

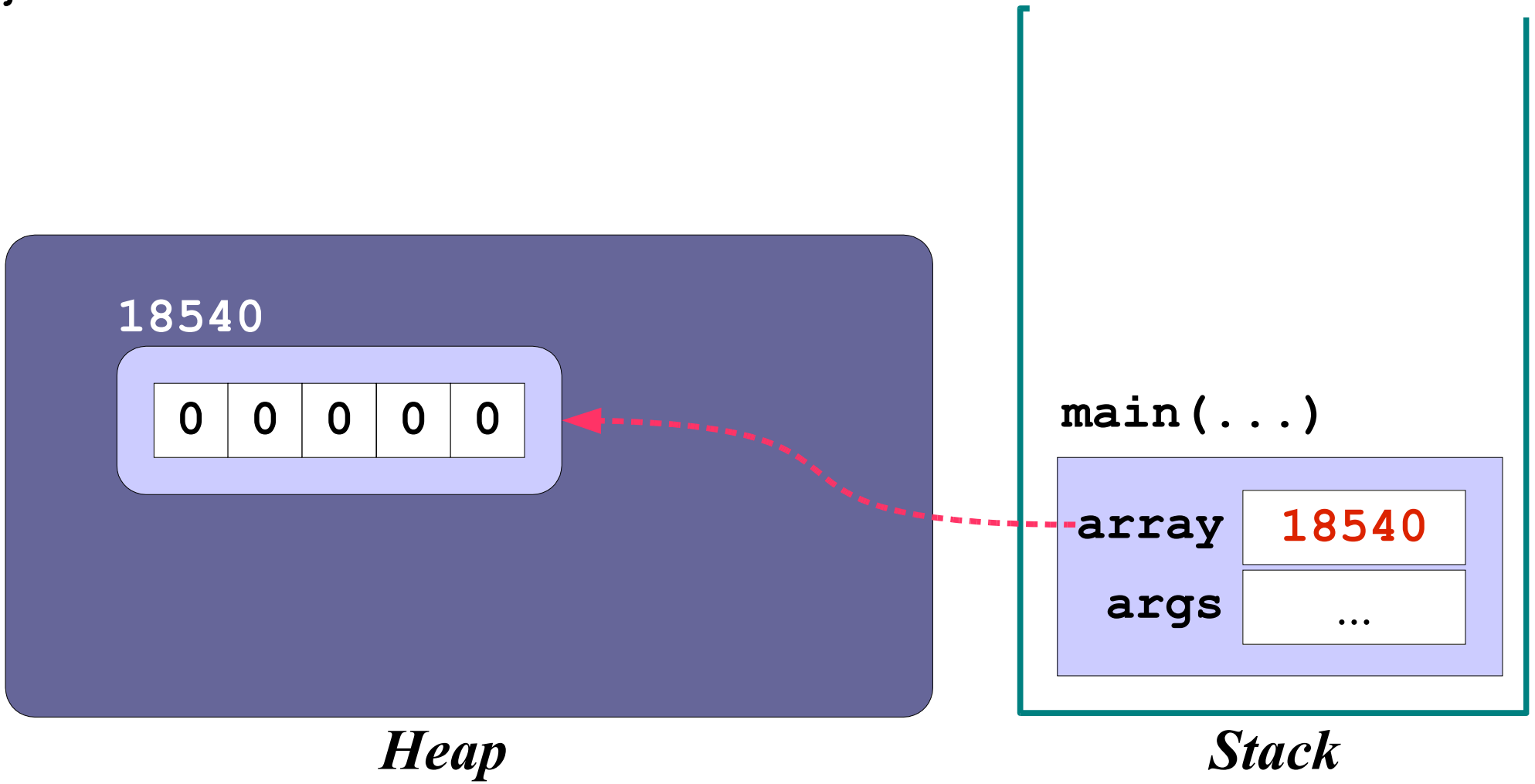
```
public class ArrayTest {
   public static void main(...) {
      int[] array;
      array = new int[5];
      array[1] = 12;
   }
}
```

Heap



Stack

```
public class ArrayTest {
   public static void main(...) {
      int[] array;
      array = new int[5];
      array[1] = 8;
   }
}
```



```
public class ArrayTest {
   public static void main(...) {
      int[] array;
      array = new int[5];
      array[1] = 8;
   }
}
```