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);
                                      main(...)
                                           a
                                       args
                                           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);
                                     main(...)
                                       args
```

```
public class Test1 {
  public static void f(int x) {
     int y = 1;
     y = x + y;
     int z = 10;
     x = y * z;
                                      f(2)
  public static void main(...) {
     int a = 2;
                                           Z
     f(a);
                                           У
     System.out.println(a);
                                           X
                                      main(...)
                                           a
                                        args
```

Stack

```
public class Test1 {
  public static void f(int x) {
    int y = 1;
     y = x + y;
     int z = 10;
     x = y * z;
                                      f(2)
  public static void main(...) {
     int a = 2;
                                           Z
     f(a);
                                           У
     System.out.println(a);
                                           X
                                      main(...)
                                           a
                                        args
```

```
public class Test1 {
  public static void f(int x) {
     int y = 1;
    y = x + y;
     int z = 10;
     x = y * z;
                                      f(2)
  public static void main(...) {
     int a = 2;
                                           Z
     f(a);
                                           У
     System.out.println(a);
                                           X
                                      main(...)
                                           a
                                        args
```

```
public class Test1 {
  public static void f(int x) {
     int y = 1;
     y = x + y;
    int z = 10;
     x = y * z;
                                      f(2)
  public static void main(...) {
                                                10
     int a = 2;
                                           Z
     f(a);
                                           У
     System.out.println(a);
                                           X
                                      main(...)
                                           a
                                        args
```

Stack

3

```
public class Test1 {
  public static void f(int x) {
     int y = 1;
     y = x + y;
     int z = 10;
    x = y * z;
                                      f(2)
  public static void main(...) {
                                                10
     int a = 2;
                                           Z
     f(a);
                                           У
     System.out.println(a);
                                                30
                                           X
                                      main(...)
                                           a
                                        args
```

Stack

3

```
public class Test1 {
  public static void f(int x) {
     int y = 1;
     y = x + y;
     int z = 10;
     x = y * z;
                                       f(2)
  public static void main(...) {
     int a = 2;
                                           Z
     f(a);
     System.out.println(a);
                                                30
                                           X
                                      main(...)
                                                 2
                                           a
                                        args
                                            Stack
```

```
public class Test1 {
  public static void f(int x) {
     int y = 1;
     y = x + y;
     int z = 10;
     x = y * z;
                                      f(2)
  public static void main(...) {
     int a = 2;
     f(a);
     System.out.println(a);
                                      main(...)
```

10

30

2

У

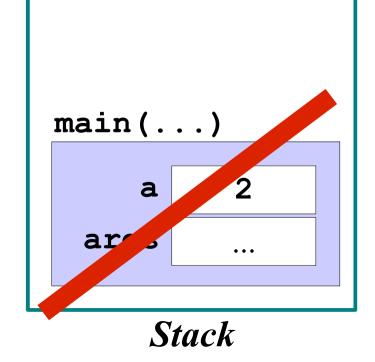
X

a

args

```
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);
                                     main(...)
                                       args
```

```
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);
```

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);
    }
}
```

```
main(...)

b ?

lt ?

args ...
```

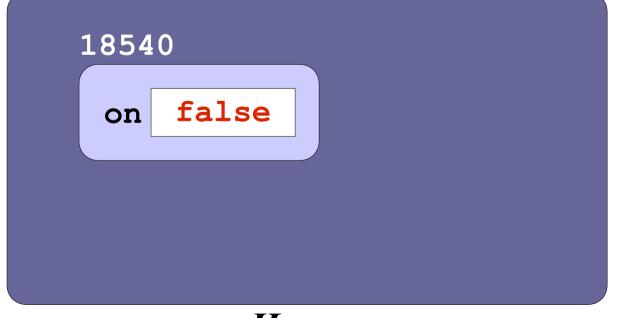
Stack

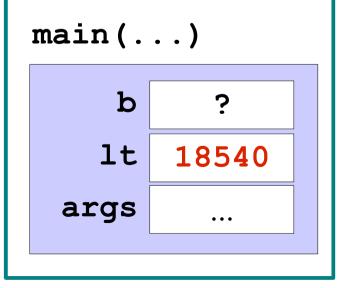
```
public class LightProg {
   public static void main(...) {
     Light lt;
     lt = new Light();
     lt.switchOn();
     boolean b;
     b = lt.isOn();
     System.out.println(b);
   }
}
```

main()	
b	
1t	?
args	•••

Stack

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





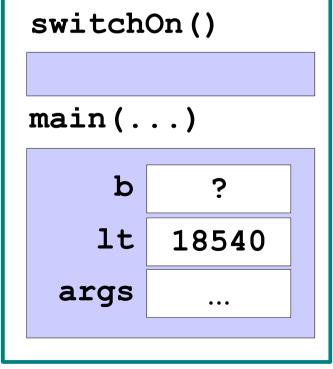
Heap

Stack

```
public class LightProg {
  public static void main(...) {
     Light 1t;
     lt = new Light();
     lt.switchOn();
     boolean b;
     b = lt.isOn();
     System.out.println(b);
    18540
                                      main(...)
         false
     on
                                           b
                                              18540
                                          ·1t
                                        args
                                           Stack
             Heap
```

```
public class LightProg {
  public static void main(...) {
     Light 1t;
     lt = new Light();
     lt.switchOn();
     boolean b;
     b = lt.isOn();
     System.out.println(b);
```



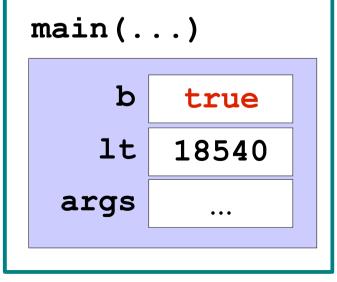


Heap

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

Generating Garbage

```
public class LightProg {
  public static void main(...) {
     Light 1t;
     lt = new Light();
     lt.switchOn();
     lt = new Light();
     System.out.println(lt.isOn());
    18540
                                      main(...)
          true
     on
                                         -lt
                                              18540
                                        args
                                           Stack
             Heap
```

```
public class LightProg {
  public static void main(...) {
     Light 1t;
     lt = new Light();
     lt.switchOn();
     lt = new Light();
     System.out.println(lt.isOn());
    18540
          true
     on
                                      main(...)
              18560
                                         -lt
                                              18560
                                        args
                   false
               on
                                           Stack
             Heap
```

```
public class LightProg {
  public static void main(...) {
     Light 1t;
     lt = new Light();
     lt.switchOn();
     lt = new Light();
     System.out.println(lt.isOn());
    18540
          true
     on
                                      main(...)
              18560
                                         -lt
                                              18560
                                        args
                   false
               on
                                           Stack
             Heap
```

Aliasing

```
public class LightProg {
  public static void main(...) {
     Light 1t;
     lt = new Light();
     lt.switchOn();
     Light 1t2 = 1t;
     lt2.switchOff();
     System.out.println(lt.isOn())
    18540
                                      main(...)
          true
     on
                                        1t2
                                         -lt
                                              18540
                                       args
                                           Stack
             Heap
```

```
public class LightProg {
  public static void main(...) {
     Light 1t;
     lt = new Light();
     lt.switchOn();
     Light 1t2 = 1t;
     lt2.switchOff();
     System.out.println(lt.isOn())
    18540
                                      main(...)
          true
     on
                                        1t2
                                              18540
                                         -lt
                                              18540
                                       args
                                           Stack
             Heap
```

```
public class LightProg {
  public static void main(...) {
     Light 1t;
     lt = new Light();
     lt.switchOn();
     Light 1t2 = 1t;
     1t2.switchOff();
     System.out.println(lt.isOn())
    18540
                                      main(...)
         false
     on
                                        ·1t2
                                              18540
                                         -lt
                                              18540
                                       args
                                           Stack
             Heap
```

```
public class LightProg {
  public static void main(...) {
     Light 1t;
     lt = new Light();
     lt.switchOn();
     Light 1t2 = 1t;
     lt2.switchOff();
     System.out.println(lt.isOn())
    18540
                                      main(...)
         false
     on
                                         1t2
                                              18540
                                         -lt
                                              18540
                                        args
                                           Stack
             Heap
```

Avoiding Garbage

```
public class LightProg {
  public static void main(...) {
     Light 1t;
     lt = new Light();
     lt.switchOn();
     Light 1t2 = 1t;
     lt = new Light();
     System.out.println(lt.isOn())
    18540
                                      main(...)
          true
     on
                                        1t2
                                         -lt
                                              18540
                                       args
                                           Stack
             Heap
```

```
public class LightProg {
  public static void main(...) {
     Light 1t;
     lt = new Light();
     lt.switchOn();
     Light 1t2 = 1t;
     lt = new Light();
     System.out.println(lt.isOn())
    18540
                                      main(...)
          true
     on
                                         ·1t2
                                              18540
                                         -lt
                                              18540
                                        args
                                           Stack
             Heap
```

```
public class LightProg {
  public static void main(...) {
     Light 1t;
     lt = new Light();
     lt.switchOn();
     Light 1t2 = 1t;
     lt = new Light();
     System.out.println(lt.isOn())
    18540
                                      main(...)
          true
     on
                                        1t2
                                              18540
              18560
                                         -1t
                                              18560
                                        args
                   false
               on
                                           Stack
             Heap
```

Allocating Arrays

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



main(...)

array ?

args ...

Heap

Stack

```
public class ArrayTest {
  public static void main(...) {
     int[] array;
     array = new int[5];
     array[1] = 8;
    18540
                                      main(...)
      0
              0
                 0
                                      -array
                                              18540
                                       args
                                           Stack
             Heap
```

```
public class ArrayTest {
  public static void main(...) {
     int[] array;
     array = new int[5];
     array[1] = 8;
    18540
                                      main(...)
           0
              0
                 0
      0
                                      -array
                                              18540
                                        args
                                           Stack
             Heap
```