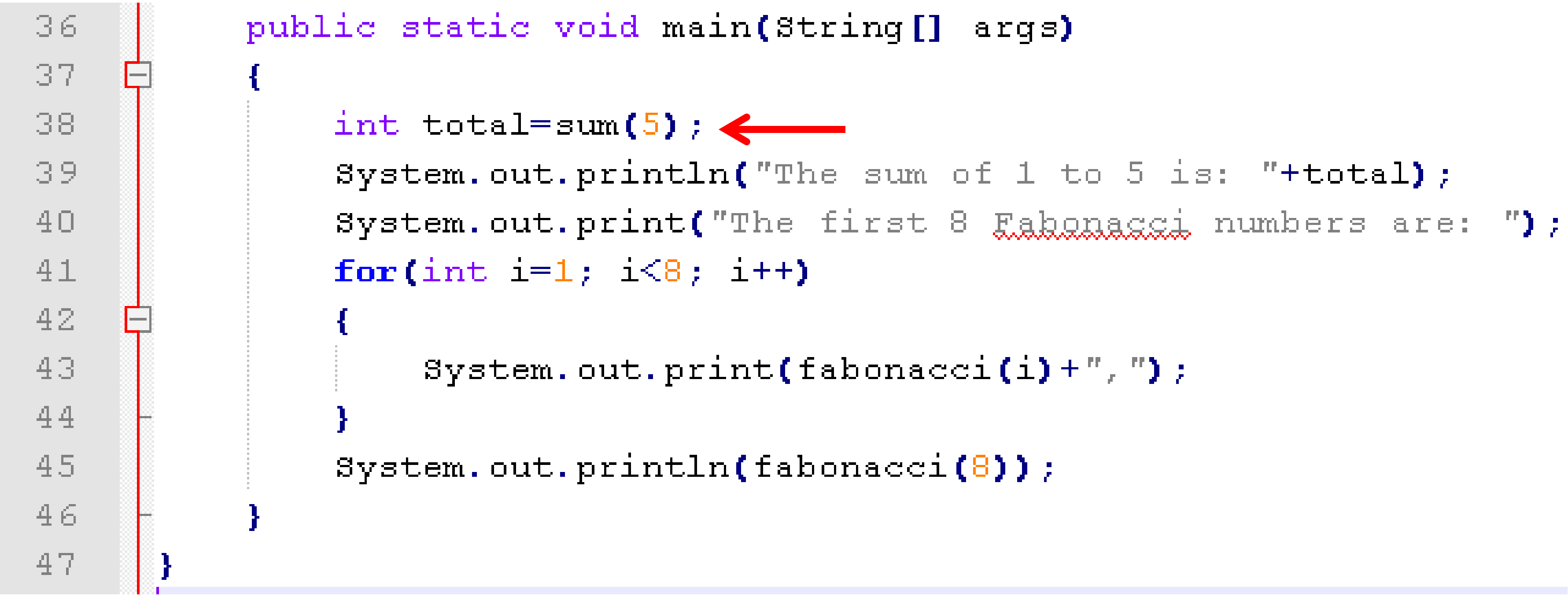
Lab 4

By Victoria Li

January 31, 2014

When run “java Lab4”, main is called.

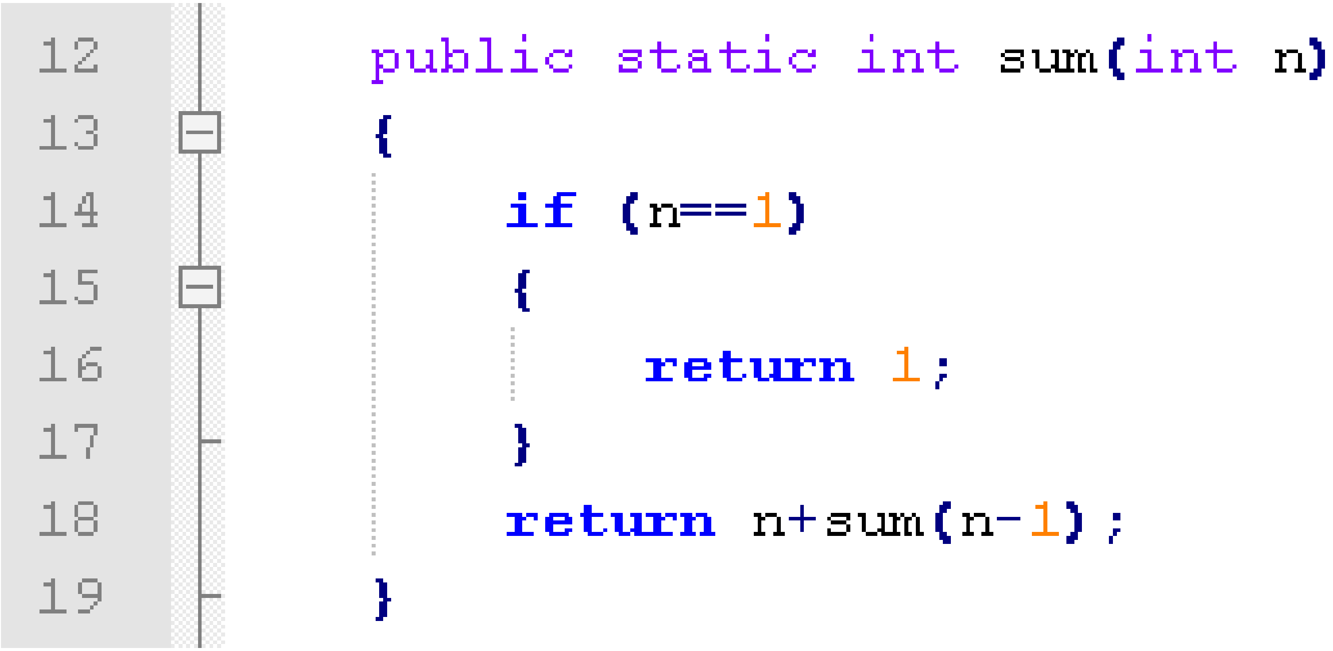
- Then line 38 is executed -> sum(5) is called.
- The program waits for the result of sum(5)
- The current content is saved on system stack



```
36      public static void main(String[] args)
37      {
38          int total=sum(5);
39          System.out.println("The sum of 1 to 5 is: "+total);
40          System.out.print("The first 8 Fabonacci numbers are: ");
41          for(int i=1; i<8; i++)
42          {
43              System.out.print(fabonacci(i)+",");
44          }
45          System.out.println(fabonacci(8));
46      }
47  }
```

Control switched to sum(5)

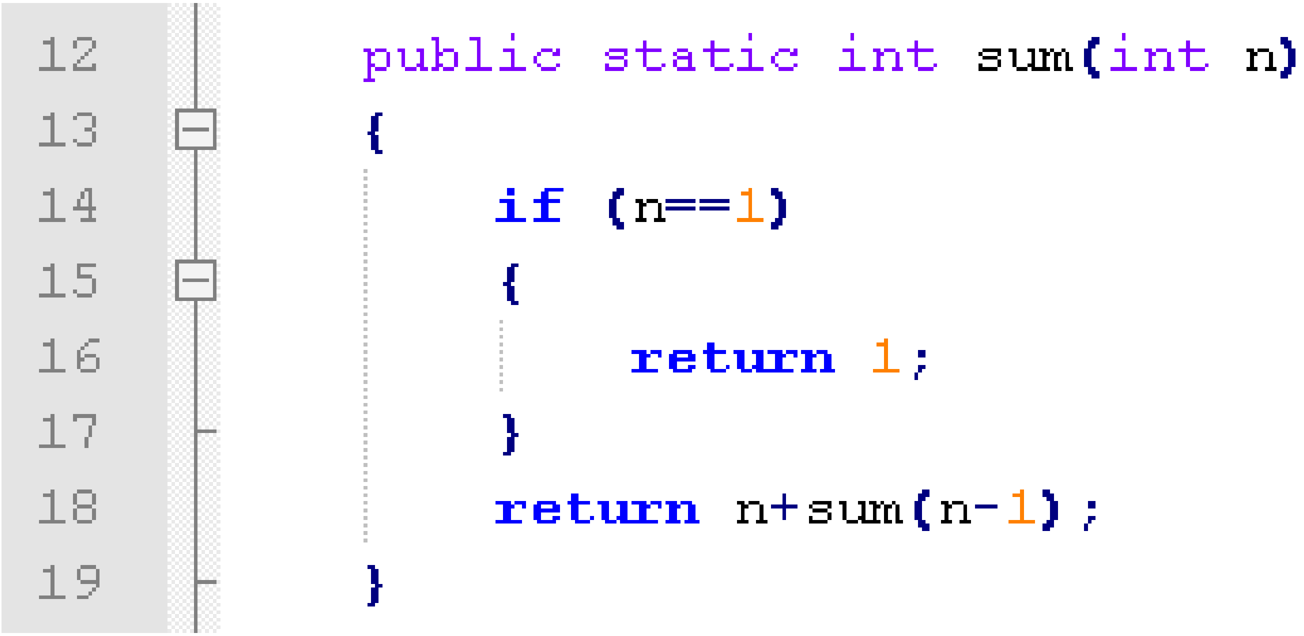
- Line 12 is executed. “int” 5 is copied to local variable n, line 14 “if (n==1)” is not true, line 18 is executed, it waits at “return 5+sum(4)”
- The current content is saved on system stack



```
12      public static int sum(int n)
13      {
14          if (n==1)
15          {
16              return 1;
17          }
18          return n+sum(n-1);
19      }
```

Control switched to sum(4)

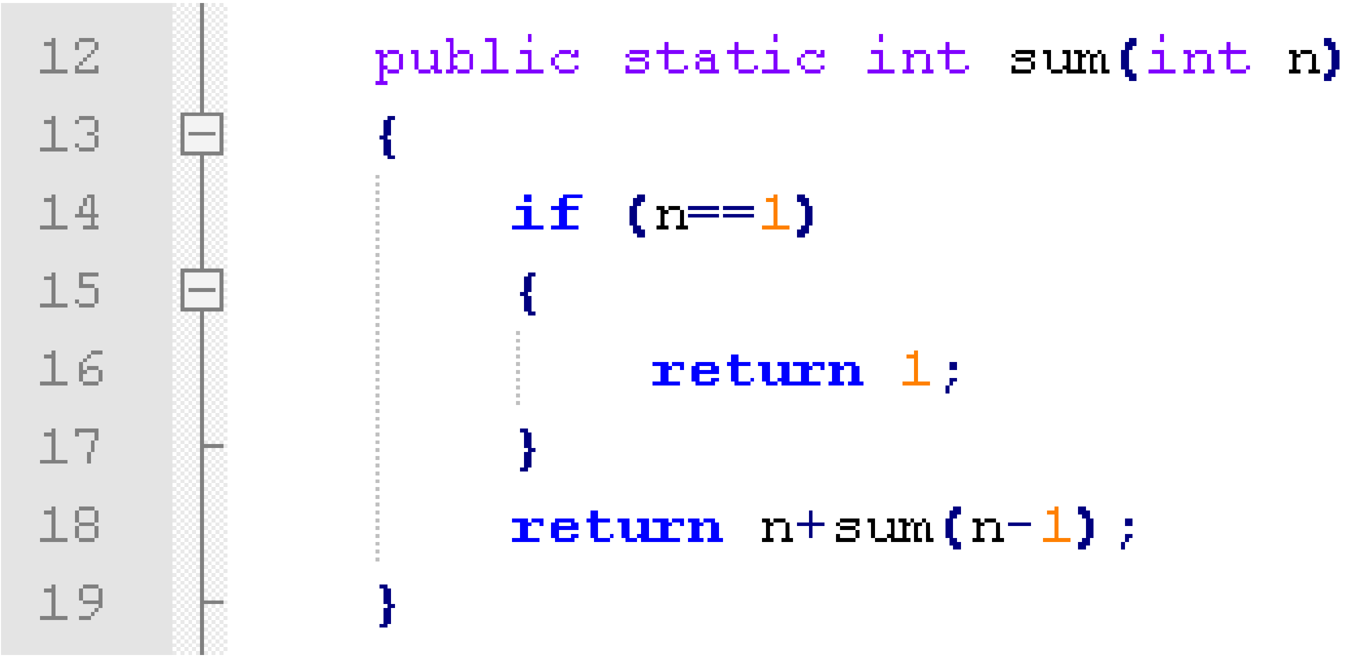
- Line 12 is executed. “int” 4 is copied to local variable n, line 14 “if (n==1)” is not true, line 18 is executed, it waits at “return 4+sum(3)”
- The current content is saved on system stack



```
12  public static int sum(int n)
13  {
14      if (n==1)
15      {
16          return 1;
17      }
18      return n+sum(n-1);
19  }
```

Control switched to sum(3)

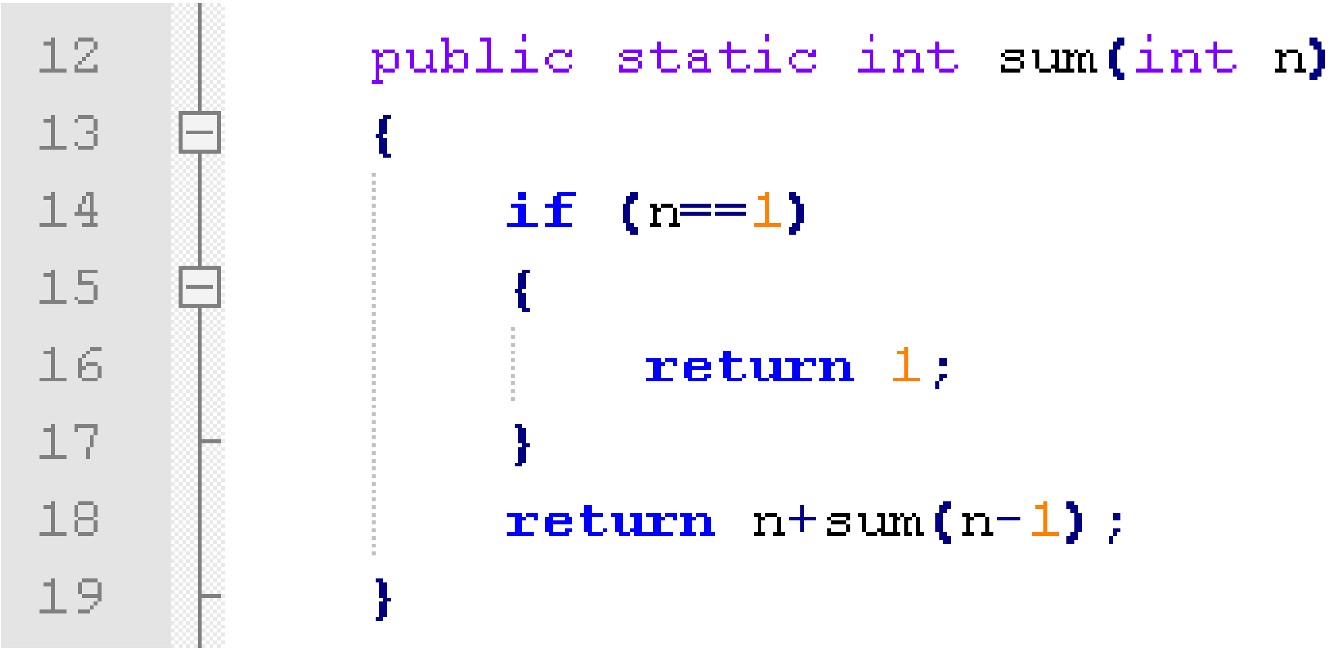
- Line 12 is executed. “int” 3 is copied to local variable n, line 14 “if (n==1)” is not true, line 18 is executed, it halts at “return 3+sum(2)”
- The current content is saved on system stack



```
12      public static int sum(int n)
13      {
14          if (n==1)
15          {
16              return 1;
17          }
18          return n+sum(n-1);
19      }
```

Control switched to sum(2)

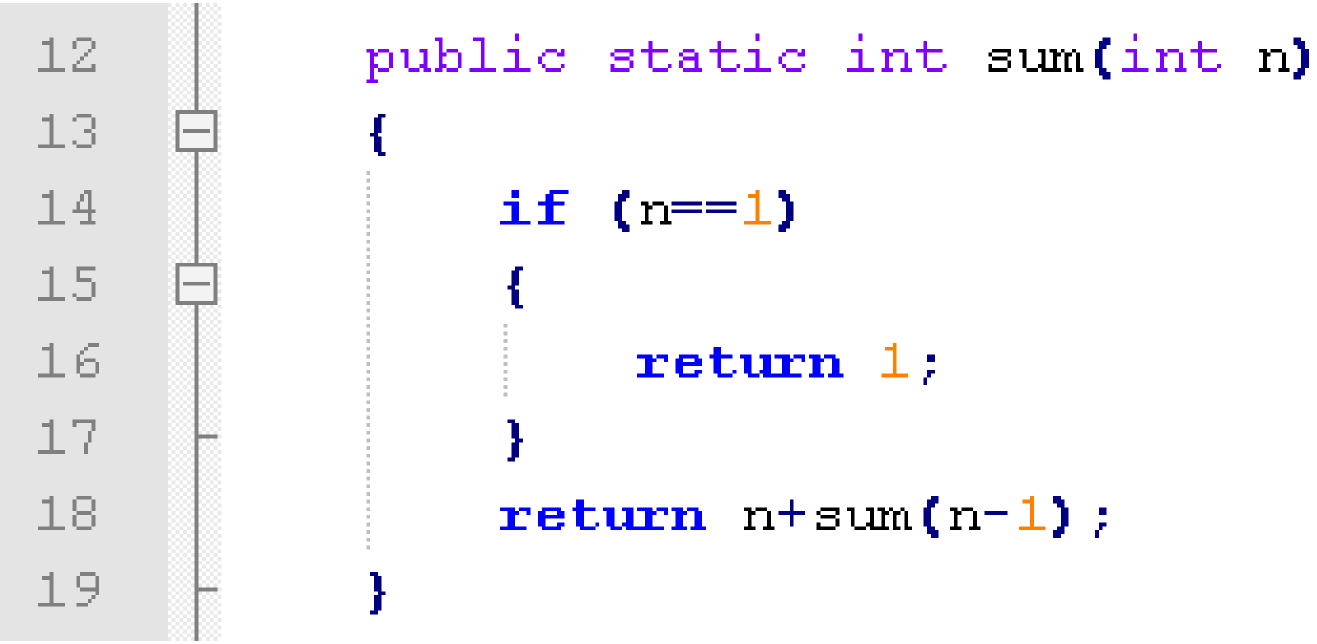
- Line 12 is executed. “int” 2 is copied to local variable n, line 14 “if (n==1)” is not true, line 18 is executed, it halts at “return 2+sum(1)”
- The current content is saved on system stack



```
12  public static int sum(int n)
13  {
14      if (n==1)
15      {
16          return 1;
17      }
18      return n+sum(n-1);
19  }
```

Control switched to sum(1)

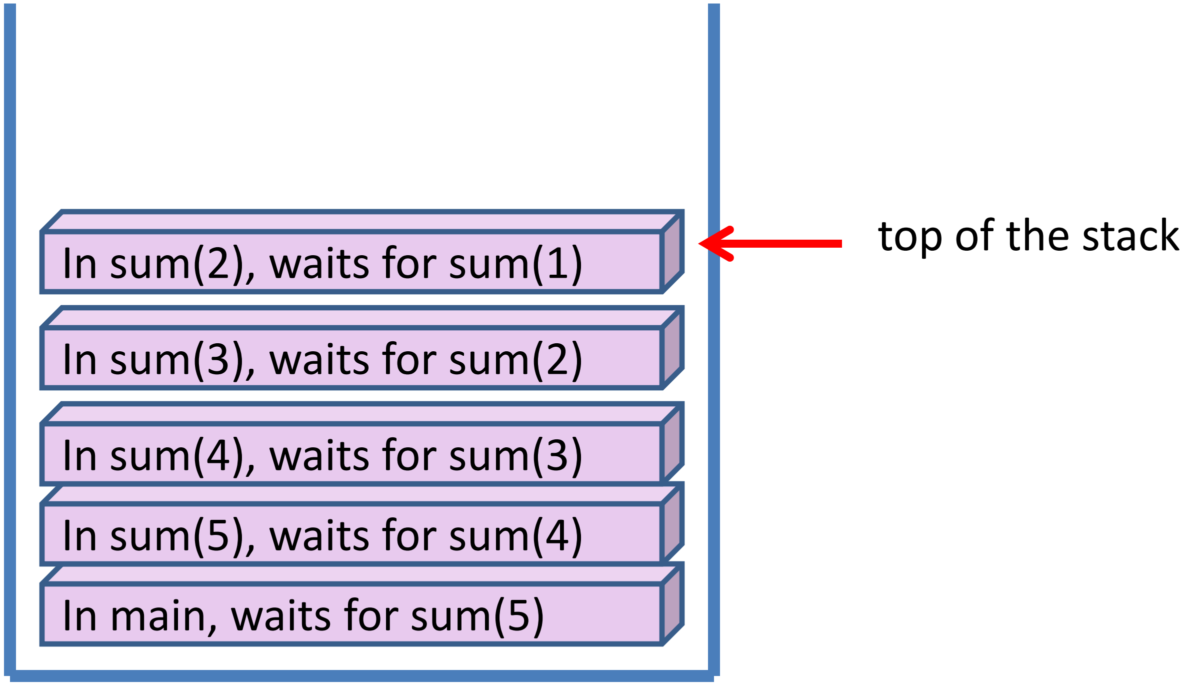
- Line 12 is executed. “int” 1 is copied to local variable n, line 14 “if (n==1)” is true, line 16 is executed, it returns 1.



```
12        public static int sum(int n)
13        {
14            if (n==1)
15            {
16                return 1;
17            }
18            return n+sum(n-1);
19        }
```

Visualizing System Stack

- When control is switched because sum(x) is called, the current content is saved on system stack.



In sum(2), waits for sum(1)

In sum(3), waits for sum(2)

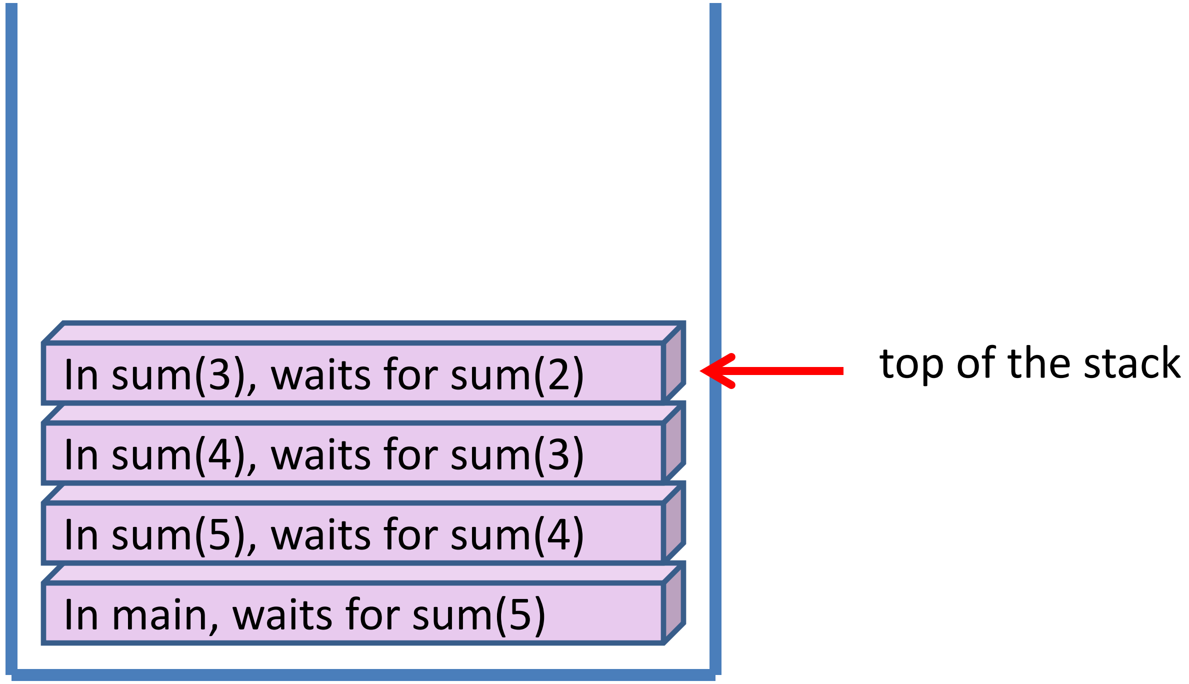
In sum(4), waits for sum(3)

In sum(5), waits for sum(4)

In main, waits for sum(5)

top of the stack

- When sum(1) is returned, Control is switched back to Line 18 of sum(2) by popping it from the system stack
- return 2+1;



In sum(3), waits for sum(2)

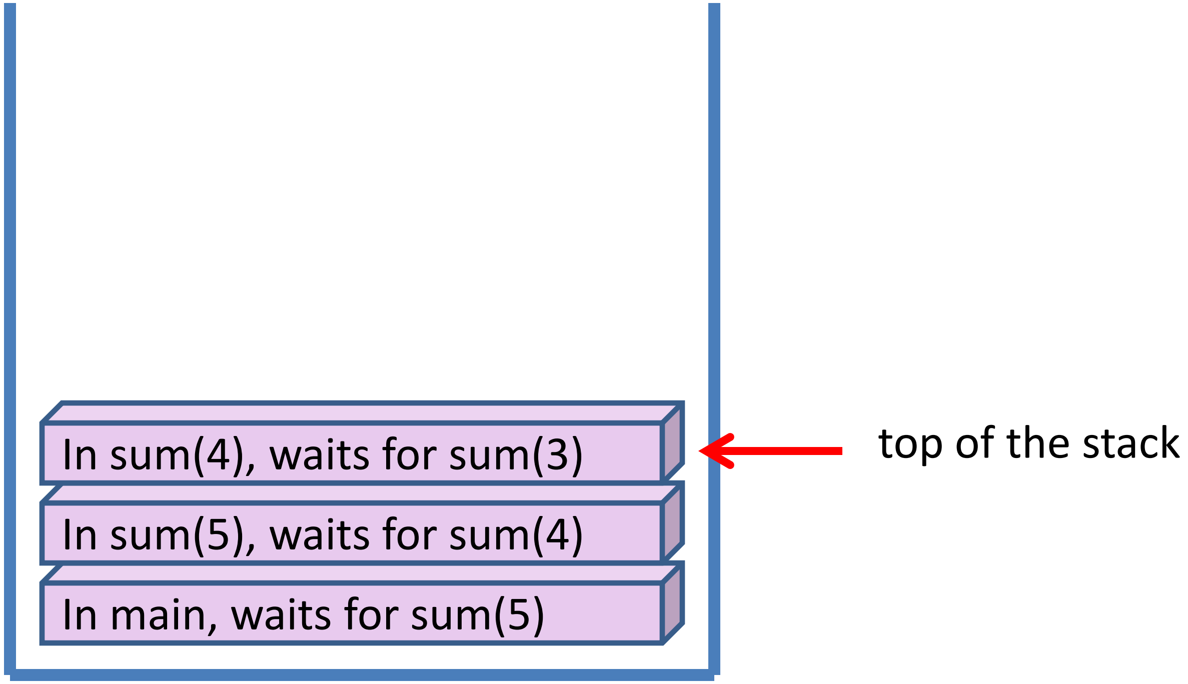
In sum(4), waits for sum(3)

In sum(5), waits for sum(4)

In main, waits for sum(5)

top of the stack

- When sum(2) is returned, Control is switched back to Line 18 of sum(3) by popping it from the system stack
- return 3+3;



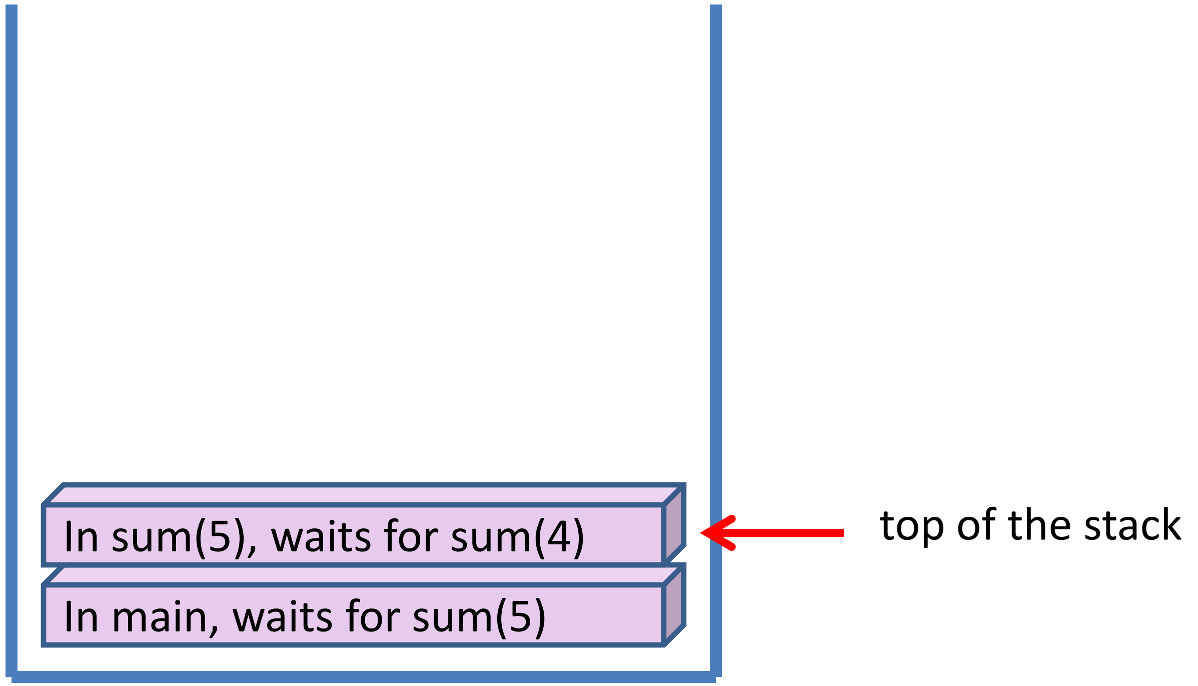
In sum(4), waits for sum(3)

In sum(5), waits for sum(4)

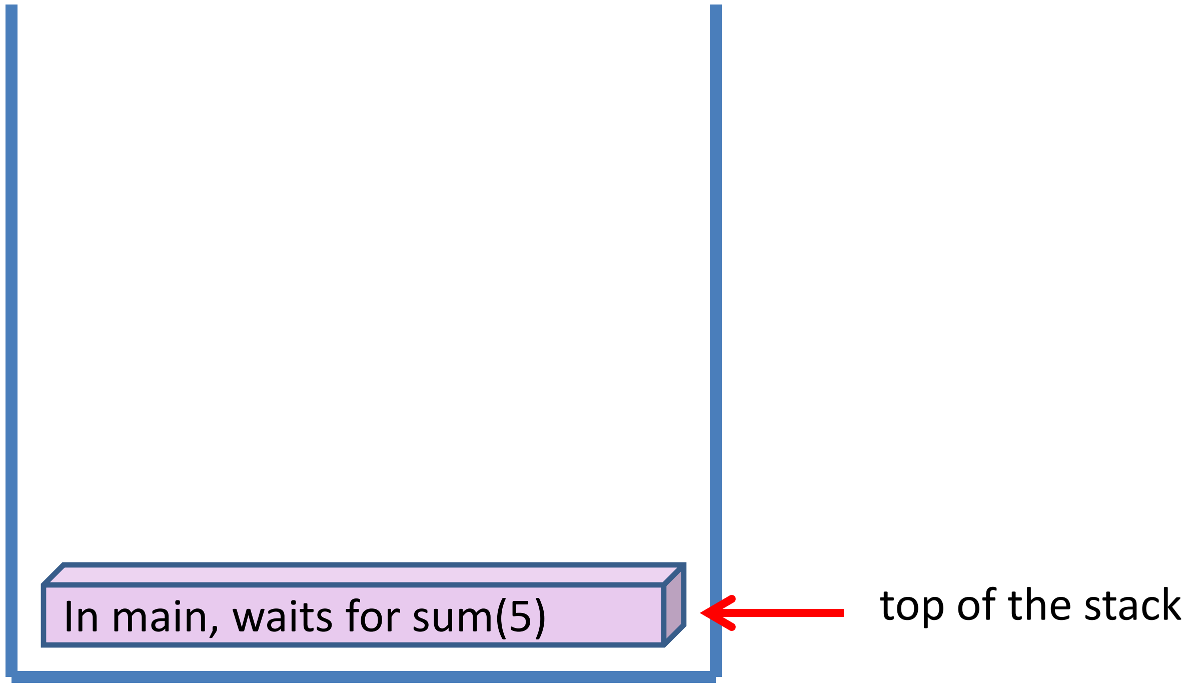
In main, waits for sum(5)

top of the stack

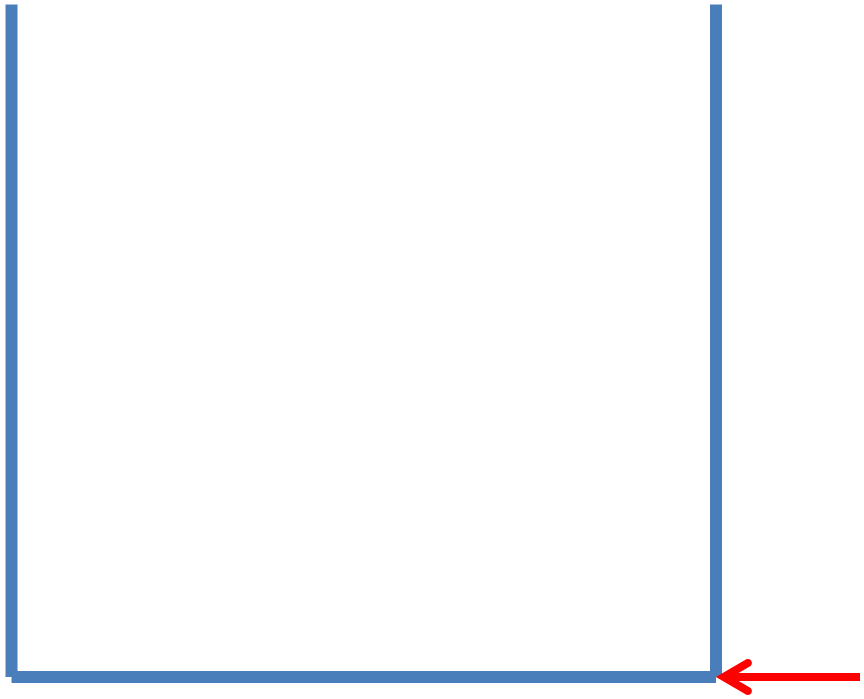
- When sum(3) is returned, Control is switched back to Line 18 of sum(4) by popping it from the system stack
- return 4+6;



- When sum(4) is returned, Control is switched back to Line 18 of sum(5) by popping it from the system stack
- return 5+10;



- When sum(5) is returned, Control is switched back to line 38 of main by popping it from the system stack
- In main, line 38, total=15;



top of the stack