

CSC180: Lecture 13

Wael Aboulsaadat

wael@cs.toronto.edu

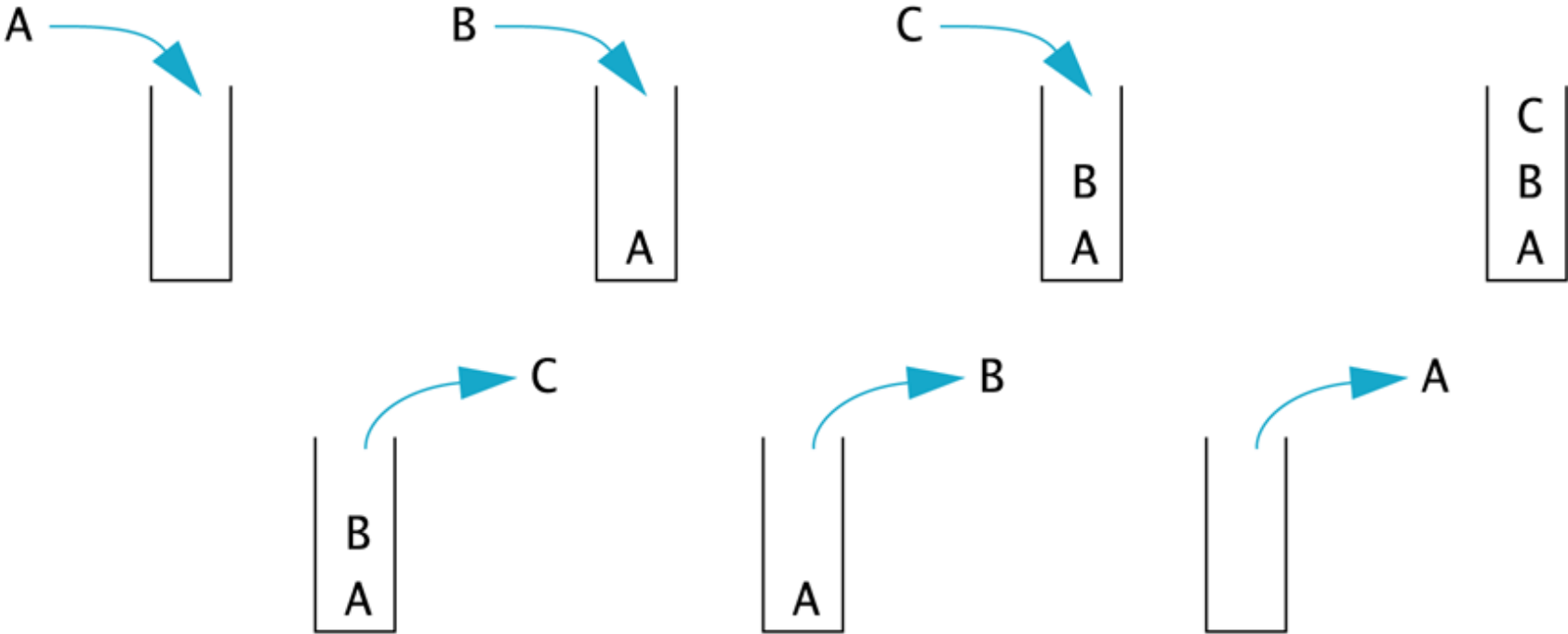
<http://portal.utoronto.ca/>

Acknowledgement: These slides are partially based on the slides supplied with Prof. Savitch book: Problem Solving with C

Functions and Call-stack

Stack

A Stack



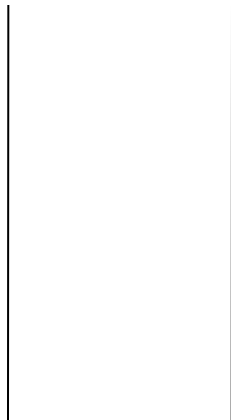
Stacks

- Computers use a structure called a stack to keep track of function calls
 - A stack is a memory structure analogous to a stack of paper
 - To place information on the stack, write it on a piece of paper and place it on top of the stack
 - To place more information on the stack, use a clean sheet of paper, write the information, and place it on the top of the stack
 - To retrieve information, only the top sheet of paper can be read, and thrown away when it is no longer needed

Last-in / First-out

- A stack is a last-in/first-out memory structure
 - The last item placed is the first that can be removed
- Whenever a function is called, the computer uses a "clean sheet of paper"
 - The function definition is copied to the paper
 - The arguments are plugged in for the parameters
 - The computer starts to execute the function body

Example



```
1  #include <stdio.h>
2
3  int f3( )
4  {
5      return 10;
6
7  }
8
9  int f2( )
10 {
11     int f3Output;
12
13     f3Output = f3( );
14
15     return f3Output * 2;
16
17 }
18
19
20 int f1()
21 {
22     int f2Output;
23
24     f2Output = f2( );
25
26     return f2Output - 5;
27 }
28
29
30 int main( )
31 {
32
33     printf(" begining of program \n");
34
35     f1( );
36
37     printf( "end of program \n" );
38
39 }
```

Example

33

```
1 #include <stdio.h>
2
3 int f3( )
4 {
5     return 10;
6
7 }
8
9 int f2( )
10 {
11     int f3Output;
12
13     f3Output = f3( );
14
15     return f3Output * 2;
16
17 }
18
19
20 int f1()
21 {
22     int f2Output;
23
24     f2Output = f2( );
25
26     return f2Output - 5;
27 }
28
29
30 int main( )
31 {
32
33     printf(" begining of program \n");
34
35     f1( );
36
37     printf( "end of program \n" );
38
39 }
```

Example

35

```
1 #include <stdio.h>
2
3 int f3( )
4 {
5     return 10;
6
7 }
8
9 int f2( )
10 {
11     int f3Output;
12
13     f3Output = f3( );
14
15     return f3Output * 2;
16
17 }
18
19
20 int f1()
21 {
22     int f2Output;
23
24     f2Output = f2( );
25
26     return f2Output - 5;
27 }
28
29
30 int main( )
31 {
32
33     printf(" begining of program \n");
34
35     f1( );
36
37     printf( "end of program \n" );
38
39 }
```


Example

35

```
1 #include <stdio.h>
2
3 int f3( )
4 {
5     return 10;
6 }
7
8
9 int f2( )
10 {
11     int f3Output;
12
13     f3Output = f3( );
14
15     return f3Output * 2;
16 }
17
18
19
20 int f1()
21 {
22     int f2Output;
23
24     f2Output = f2( );
25
26     return f2Output - 5;
27 }
28
29
30 int main( )
31 {
32
33     printf(" begining of program \n");
34
35     f1( );
36
37     printf( "end of program \n" );
38
39 }
```

Example

24

35

```
1 #include <stdio.h>
2
3 int f3( )
4 {
5     return 10;
6
7 }
8
9 int f2( )
10 {
11     int f3Output;
12
13     f3Output = f3( );
14
15     return f3Output * 2;
16
17 }
18
19
20 int f1()
21 {
22     int f2Output;
23
24     f2Output = f2( );
25
26     return f2Output - 5;
27 }
28
29
30 int main( )
31 {
32
33     printf(" begining of program \n");
34
35     f1( );
36
37     printf( "end of program \n" );
38
39 }
```

Example

24

35

```
1 #include <stdio.h>
2
3 int f3( )
4 {
5     return 10;
6
7 }
8
9 int f2( )
10 {
11     int f3Output;
12
13     f3Output = f3( );
14
15     return f3Output * 2;
16
17 }
18
19
20 int f1()
21 {
22     int f2Output;
23
24     f2Output = f2( );
25
26     return f2Output - 5;
27 }
28
29
30 int main( )
31 {
32
33     printf(" begining of program \n");
34
35     f1( );
36
37     printf( "end of program \n" );
38
39 }
```

Example

13
24
35

```
1 #include <stdio.h>
2
3 int f3( )
4 {
5     return 10;
6 }
7
8
9 int f2( )
10 {
11     int f3Output;
12
13     f3Output = f3( );
14
15     return f3Output * 2;
16 }
17
18
19
20 int f1()
21 {
22     int f2Output;
23
24     f2Output = f2( );
25
26     return f2Output - 5;
27 }
28
29
30 int main( )
31 {
32
33     printf(" begining of program \n");
34
35     f1( );
36
37     printf( "end of program \n" );
38
39 }
```

Example

13
24
35

```
1 #include <stdio.h>
2
3 int f3( )
4 {
5     return 10;
6 }
7
8
9 int f2( )
10 {
11     int f3Output;
12
13     f3Output = f3( );
14
15     return f3Output * 2;
16 }
17
18
19
20 int f1()
21 {
22     int f2Output;
23
24     f2Output = f2( );
25
26     return f2Output - 5;
27 }
28
29
30 int main( )
31 {
32
33     printf(" begining of program \n");
34
35     f1( );
36
37     printf( "end of program \n" );
38
39 }
```

Example

24

35

```
1 #include <stdio.h>
2
3 int f3( )
4 {
5     return 10;
6
7 }
8
9 int f2( )
10 {
11     int f3Output;
12
13     f3Output = f3( );
14
15     return f3Output * 2;
16
17 }
18
19
20 int f1()
21 {
22     int f2Output;
23
24     f2Output = f2( );
25
26     return f2Output - 5;
27 }
28
29
30 int main( )
31 {
32
33     printf(" begining of program \n");
34
35     f1( );
36
37     printf( "end of program \n" );
38
39 }
```

Example

35

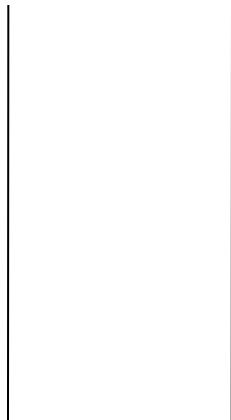
```
1 #include <stdio.h>
2
3 int f3( )
4 {
5     return 10;
6
7 }
8
9 int f2( )
10 {
11     int f3Output;
12
13     f3Output = f3( );
14
15     return f3Output * 2;
16
17 }
18
19
20 int f1()
21 {
22     int f2Output;
23
24     f2Output = f2( );
25
26     return f2Output - 5;
27 }
28
29
30 int main( )
31 {
32
33     printf(" begining of program \n");
34
35     f1( );
36
37     printf( "end of program \n" );
38
39 }
```

Example

37

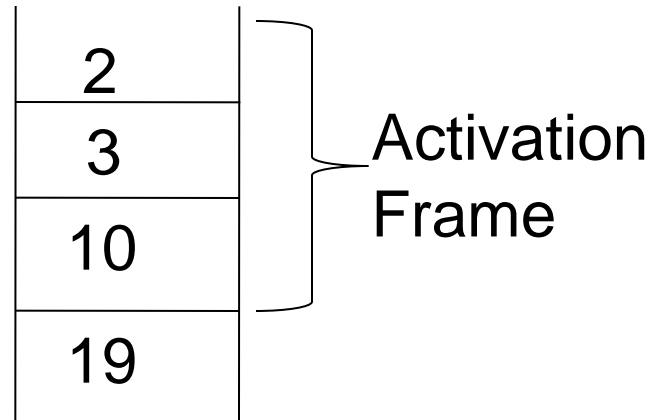
```
1 #include <stdio.h>
2
3 int f3( )
4 {
5     return 10;
6
7 }
8
9 int f2( )
10 {
11     int f3Output;
12
13     f3Output = f3( );
14
15     return f3Output * 2;
16
17 }
18
19
20 int f1()
21 {
22     int f2Output;
23
24     f2Output = f2( );
25
26     return f2Output - 5;
27 }
28
29
30 int main( )
31 {
32
33     printf(" begining of program \n");
34
35     f1( );
36
37     printf( "end of program \n" );
38
39 }
```


Example



```
1 #include <stdio.h>
2
3 int f3( )
4 {
5     return 10;
6
7 }
8
9 int f2( )
10 {
11     int f3Output;
12
13     f3Output = f3( );
14
15     return f3Output * 2;
16
17 }
18
19
20 int f1()
21 {
22     int f2Output;
23
24     f2Output = f2( );
25
26     return f2Output - 5;
27 }
28
29
30 int main( )
31 {
32
33     printf(" begining of program \n");
34
35     f1( );
36
37     printf( "end of program \n" );
38
39 }
```

Activation Frame



```
1  #include <stdio.h>
2
3  int power( int nValue, int nPower )
4  {
5  → int nProduct, nIndex;
6
7     nProduct = 1;
8     for( nIndex = 0; nIndex < nPower; nIndex = nIndex + 1 )
9         nProduct = nProduct * nValue;
10
11    return nProduct;
12 }
13
14
15 int main( )
16 {
17     int nX = 10, nExp = 3, nResult;
18
19     nResult = power( nX , nExp );
20
21     printf( "Result = %d \r\n", nResult );
22 }
23
24
```

Recall: variable declaration yields in memory space allocation

```
1  #include <stdio.h>
2
3  int power( int nValue, int nPower )
4  {
5  → int nProduct, nIndex;
6
7     nProduct = 1;
8     for( nIndex = 0; nIndex < nPower; nIndex = nIndex + 1 )
9         nProduct = nProduct * nValue;
10
11    return nProduct;
12 }
13
14
15 int main( )
16 {
17     int nX = 10, nExp = 3, nResult;
18
19     nResult = power( nX , nExp );
20
21     printf( "Result = %d \r\n", nResult );
22
23     nResult = power( nX, nExp );
24
25     printf( "Result = %d \r\n", nResult );
26 }
27
28
```