This is CS 50.

数

Harvard College's Introduction to Computer Science I

COMPUTER SCIENCE 50

WEEK 5

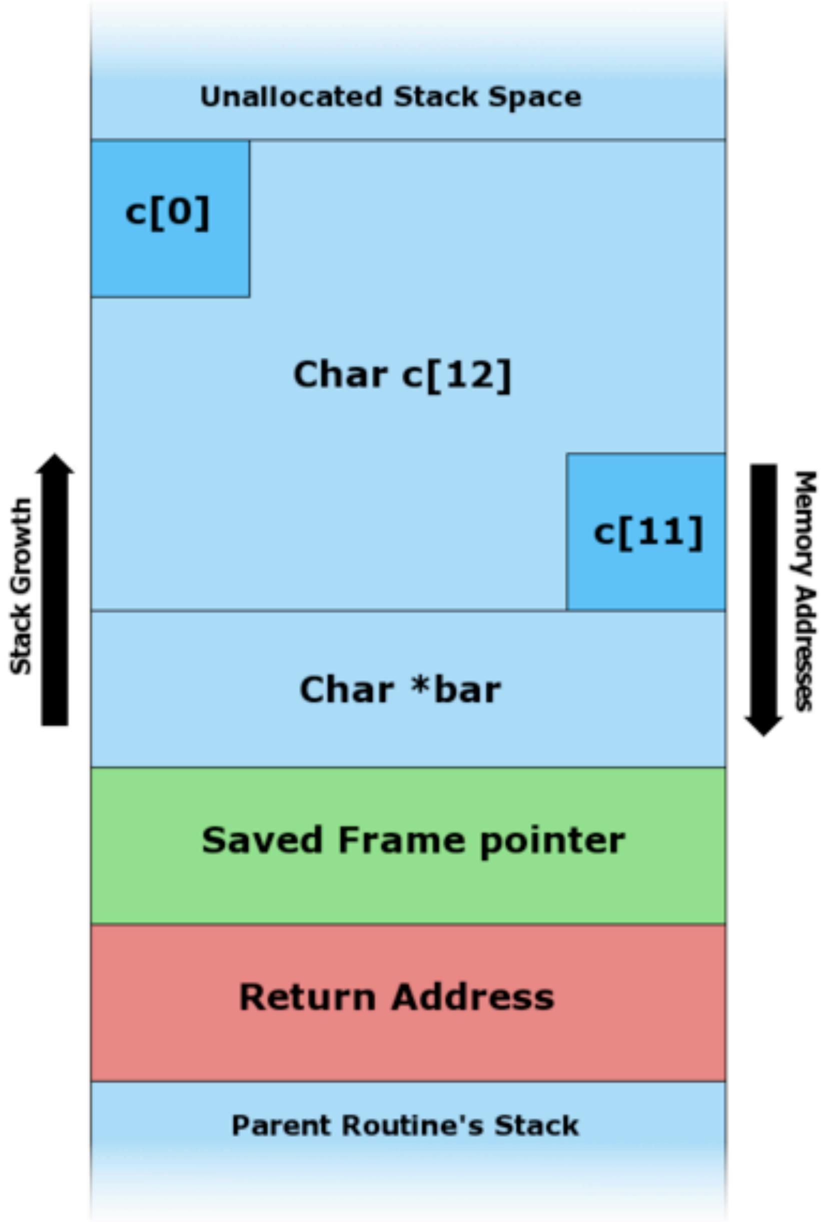
DAVID J. MALAN '99
malan@post.harvard.edu

Buffer Overflow Attacks

```
#include <string.h>

void foo (char *bar)
{
   char  c[12];

   memcpy(c, bar, strlen(bar));  // no bounds checking...
}

int main (int argc, char **argv)
{
   foo(argv[1]);
}
```

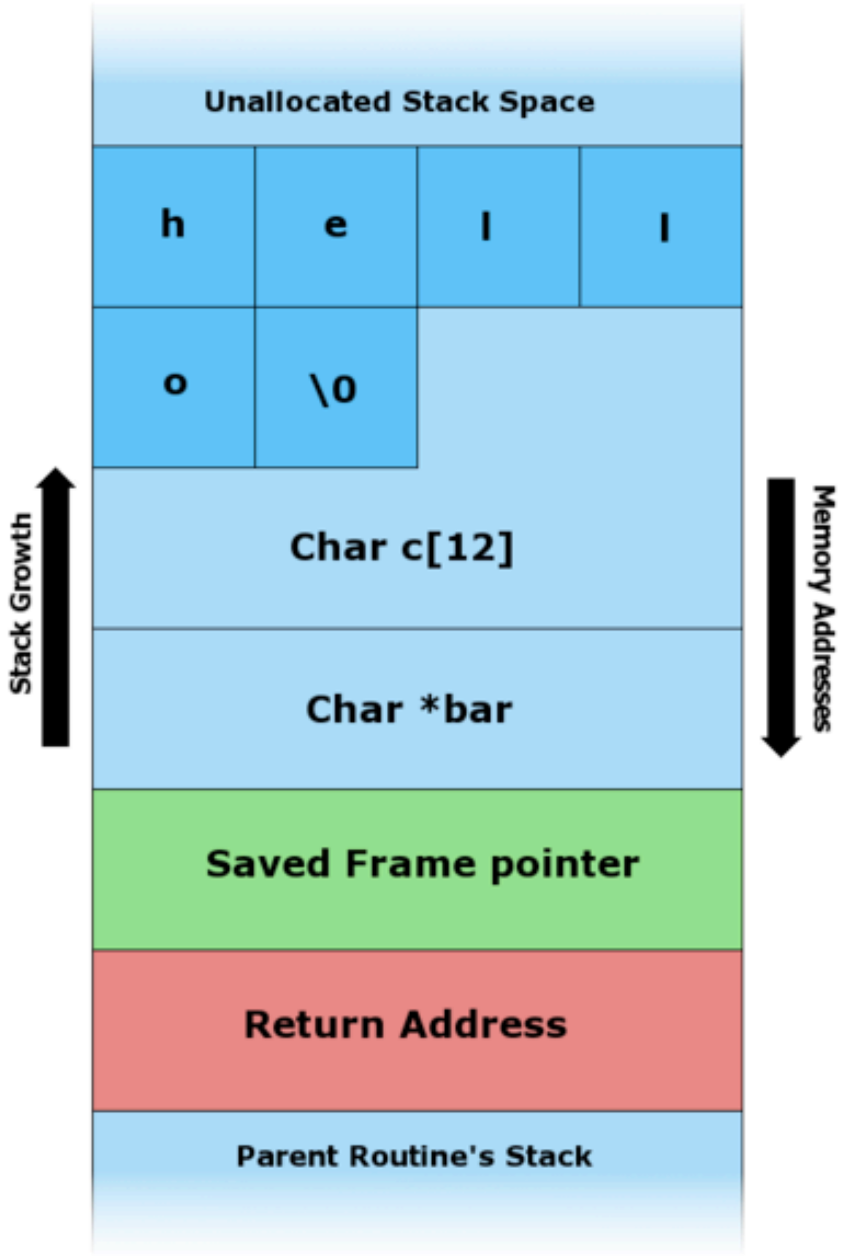
Example excerpted from http://en.wikipedia.org/wiki/Stack_buffer_overflow.

Buffer Overflow Attacks



Example excerpted from http://en.wikipedia.org/wiki/Stack_buffer_overflow.

Buffer Overflow Attacks



Unallocated Stack Space

h e l l

o \0

Char c[12]

Char *bar

Saved Frame pointer

Return Address

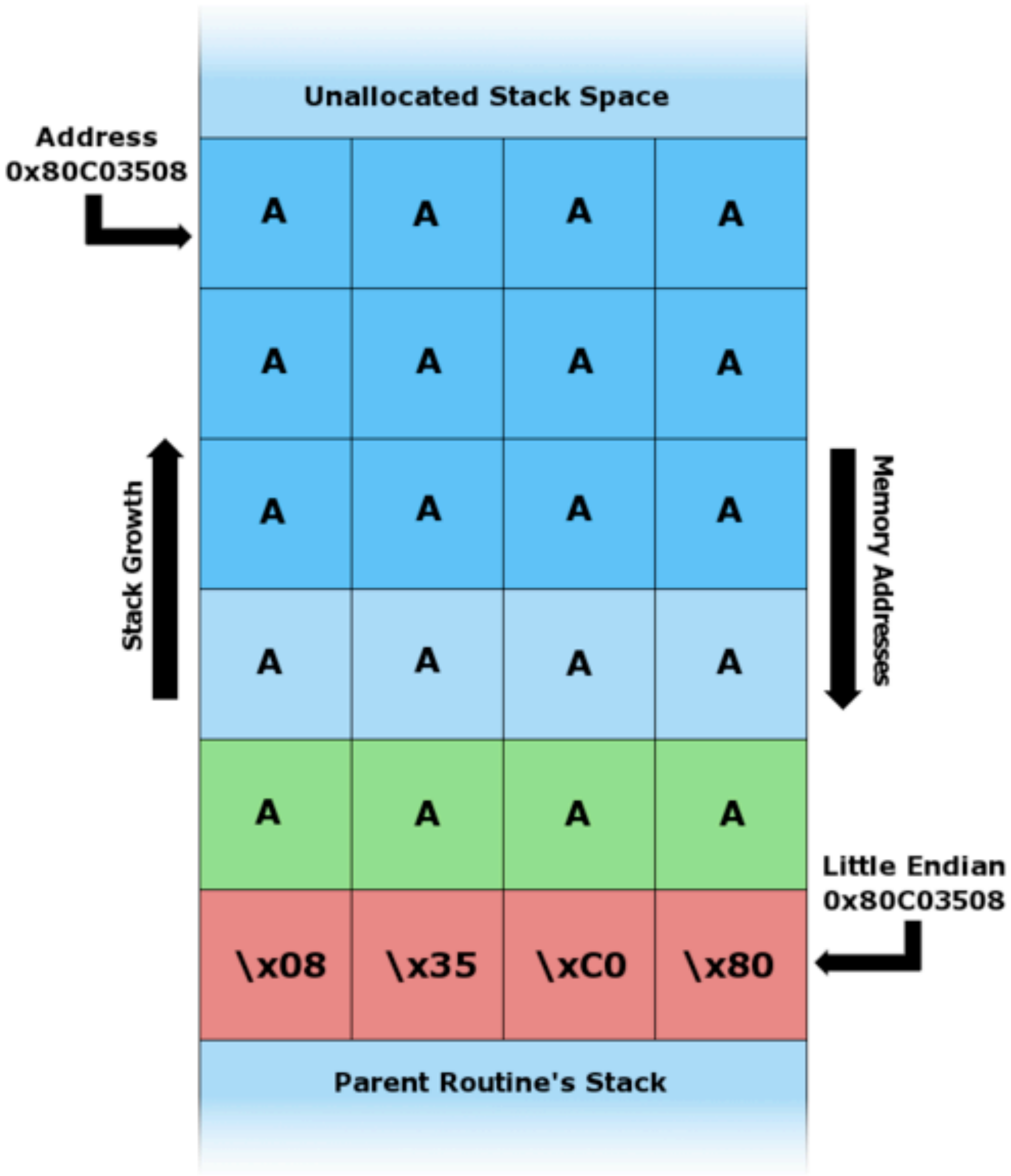
Parent Routine's Stack

Stack Growth

Memory Addresses

Example excerpted from http://en.wikipedia.org/wiki/Stack_buffer_overflow.

Buffer Overflow Attacks



Unallocated Stack Space

Address 0x80C03508

A	A	A	A
A	A	A	A
A	A	A	A
A	A	A	A
A	A	A	A
\x08	\x35	\xC0	\x80

Stack Growth

Memory Addresses

Little Endian 0x80C03508

Parent Routine's Stack

Example excerpted from http://en.wikipedia.org/wiki/Stack_buffer_overflow.

NOP Sleds

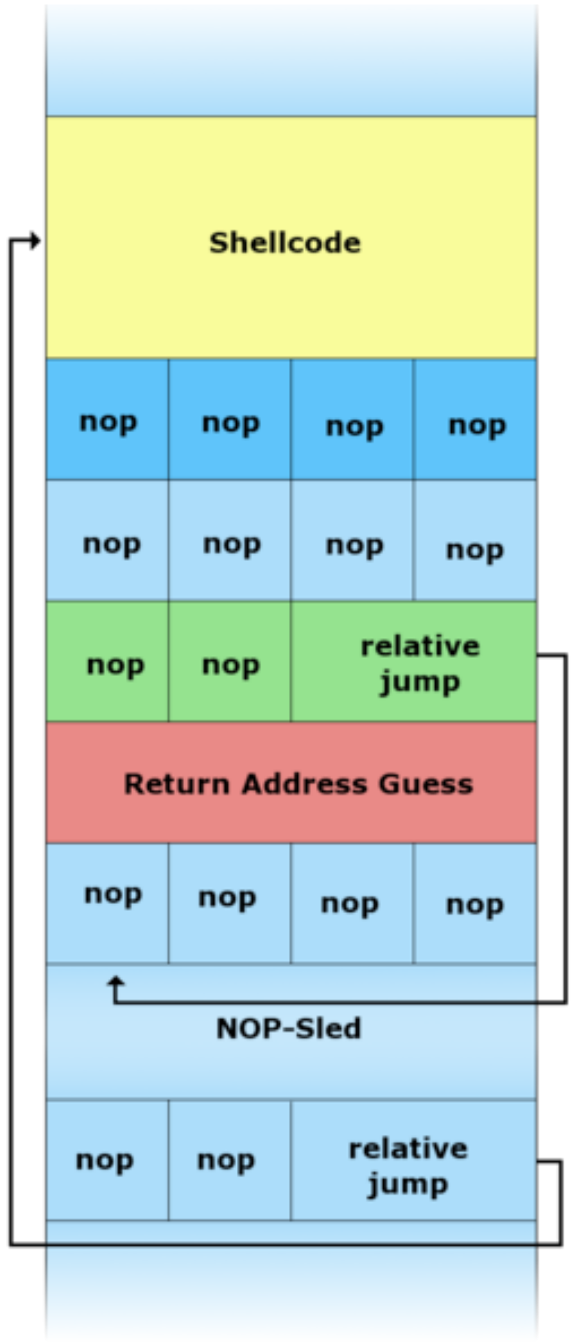


Figure excerpted from http://en.wikipedia.org/wiki/Buffer_overflow.

struct

(and header files)

```
typedef struct
{
    int id;
    char *name;
    char *house;
}
student;
```

see
`structs.h, structs1.c`

File I/O

```
fopen/fclose

fscanf/fprintf

fread/fwrite

feof

...
```

see
structs.h, structs2.c

CSI:
CRIME SCENE INVESTIGATION

Singly Linked Lists

```
typedef struct node
{
    int n;
    struct node *next;
}
node;
```

see
`list1.{c,h}`

Singly Linked Lists

```
typedef struct
{
    int id;
    char *name;
    char *house;
}
student;

typedef struct node
{
    student *student;
    struct node *next;
}
node;
```

see
`list2.{c,h}`

Singly Linked Lists

Representation

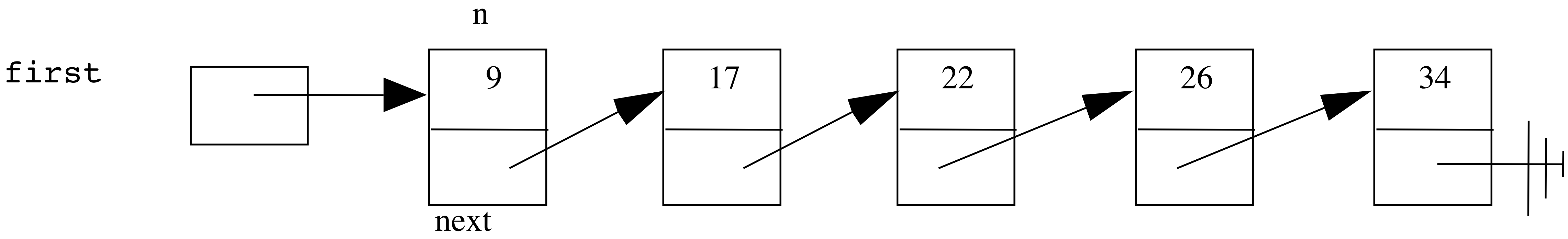


Figure adapted from http://cs.calvin.edu/books/c++/ds/1e/.

Singly Linked Lists

Traversal

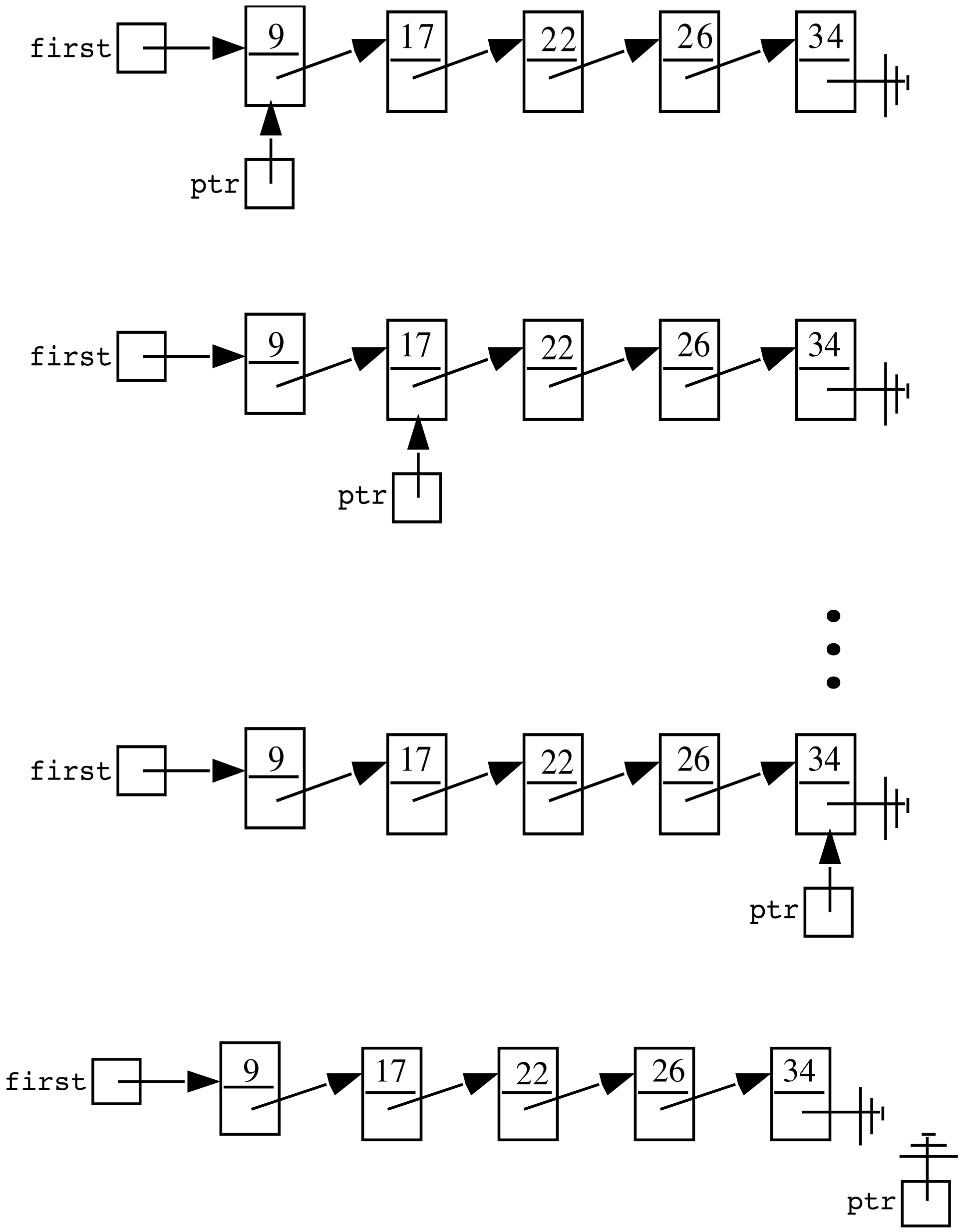


Figure adapted from http://cs.calvin.edu/books/c++/ds/1e/.

Singly Linked Lists

Insertion in Middle: Step 1

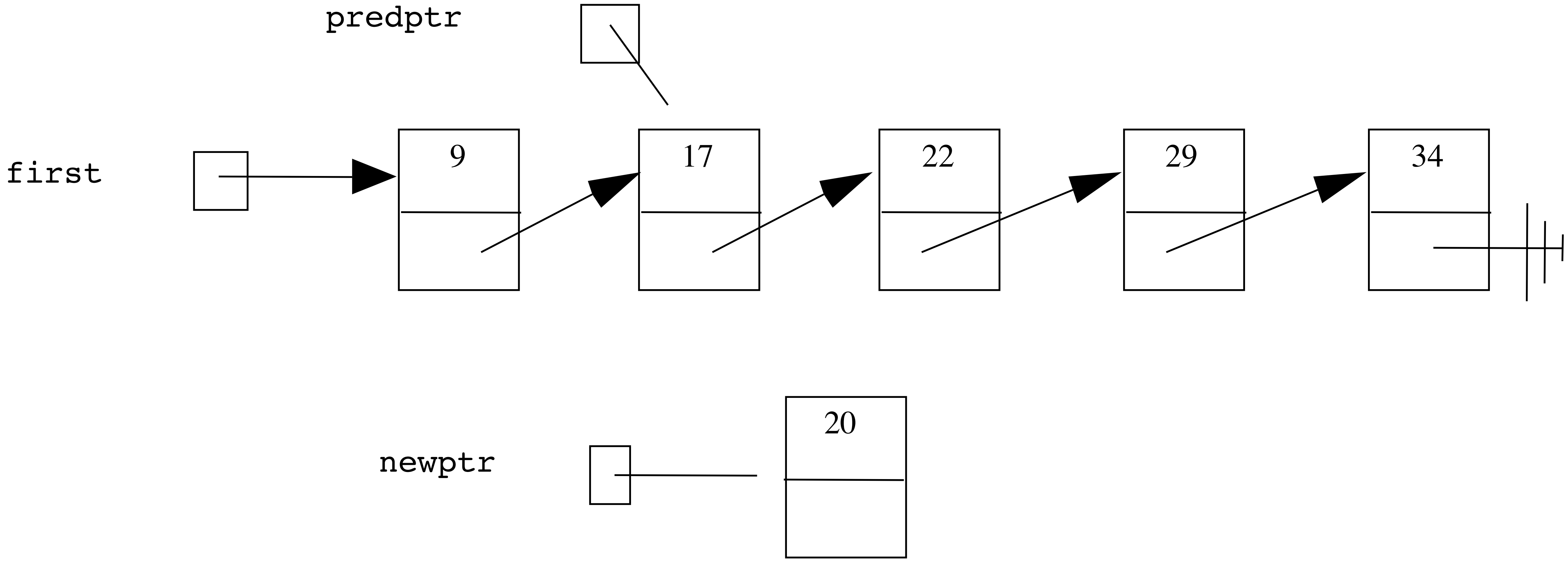


Figure adapted from http://cs.calvin.edu/books/c++/ds/1e/.

Singly Linked Lists

Insertion in Middle: Step 2

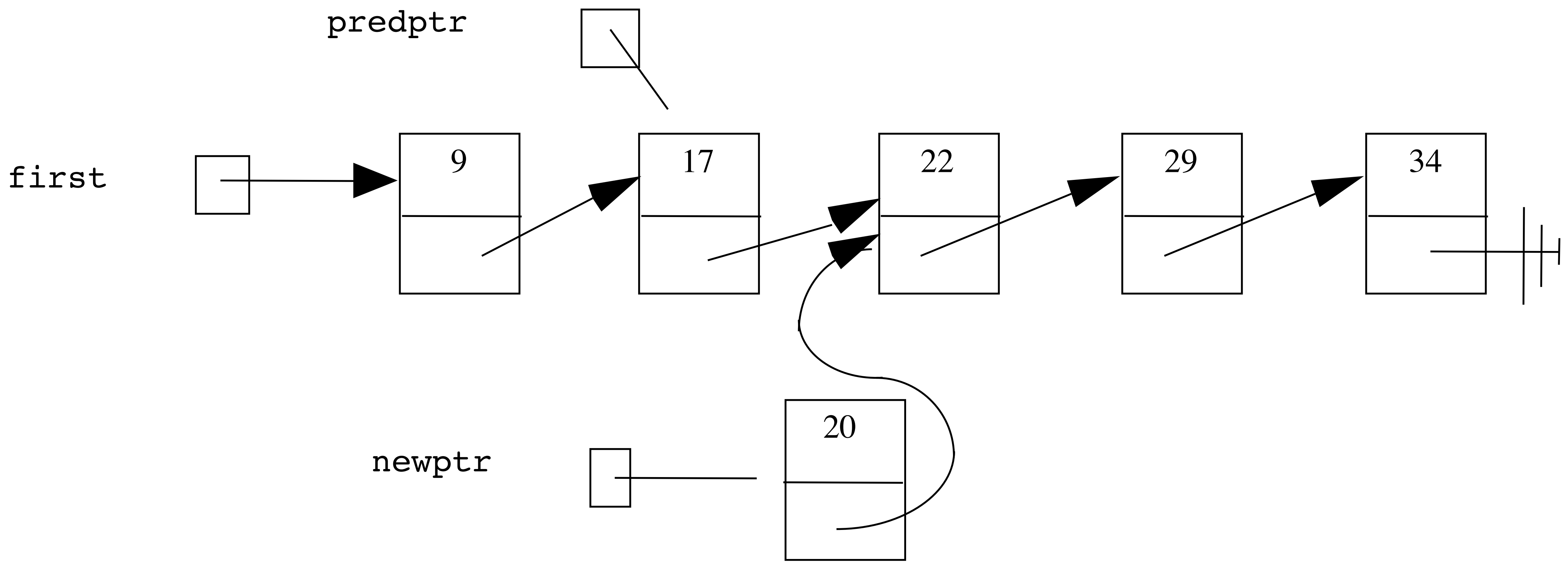


Figure adapted from http://cs.calvin.edu/books/c++/ds/1e/.

Singly Linked Lists

Insertion in Middle: Step 3

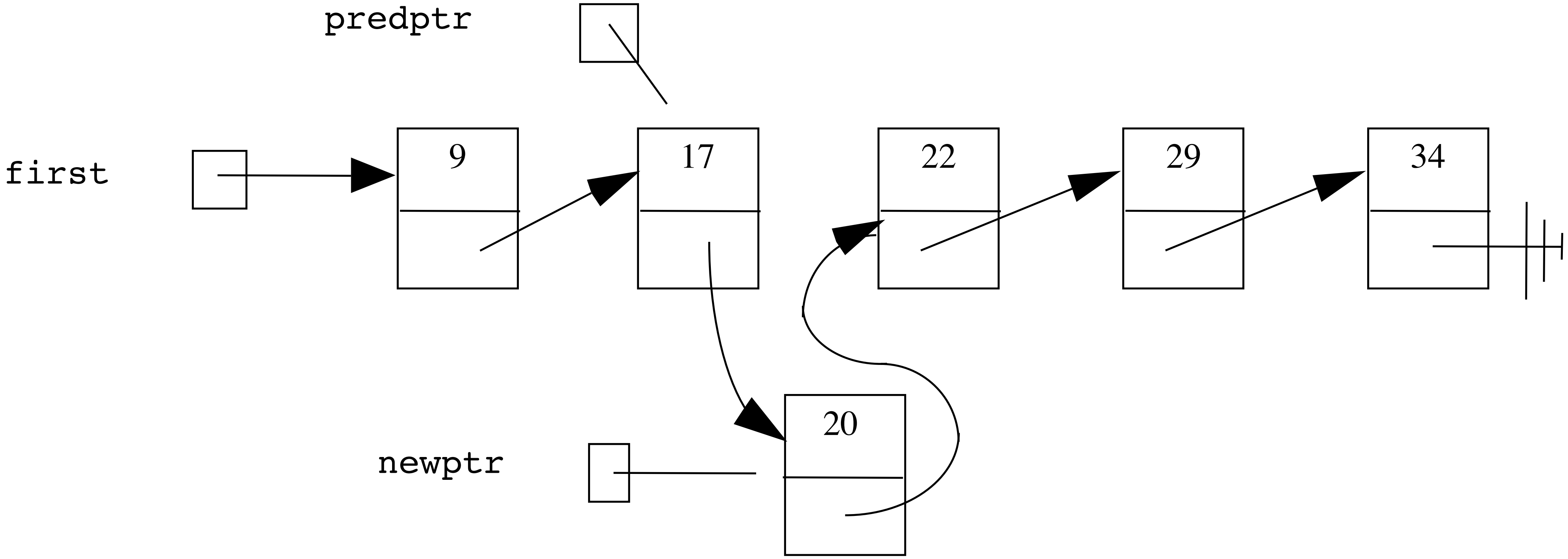


Figure adapted from http://cs.calvin.edu/books/c++/ds/1e/.

Singly Linked Lists

Insertion at Tail

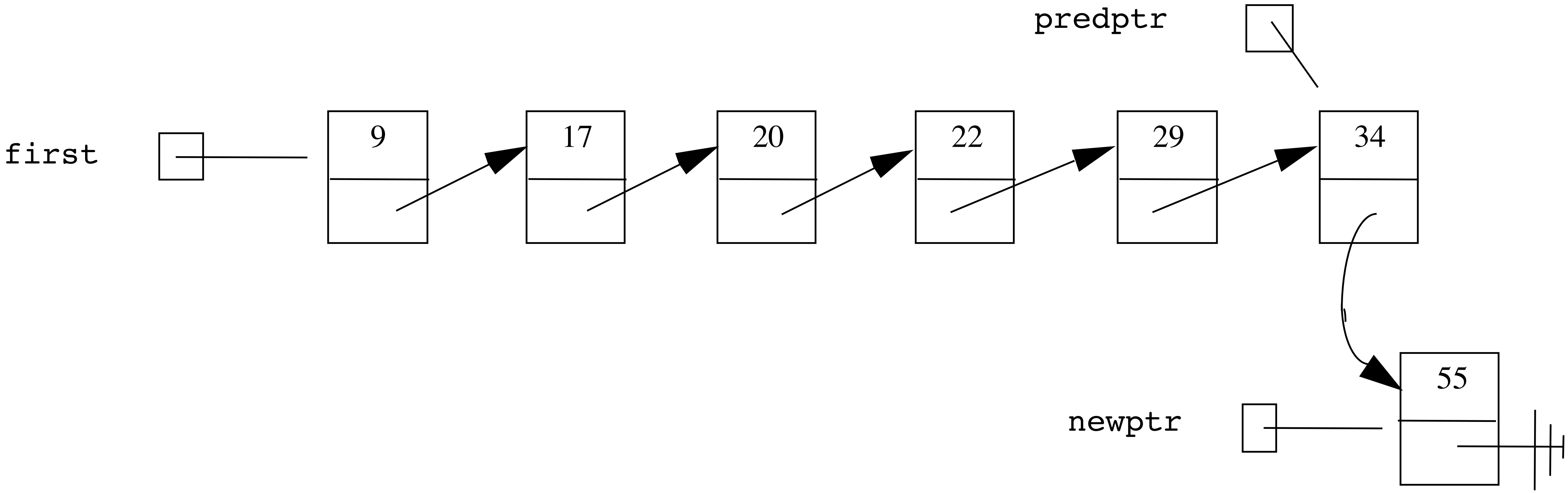


Figure adapted from http://cs.calvin.edu/books/c++/ds/1e/.

Singly Linked Lists

Insertion at Head

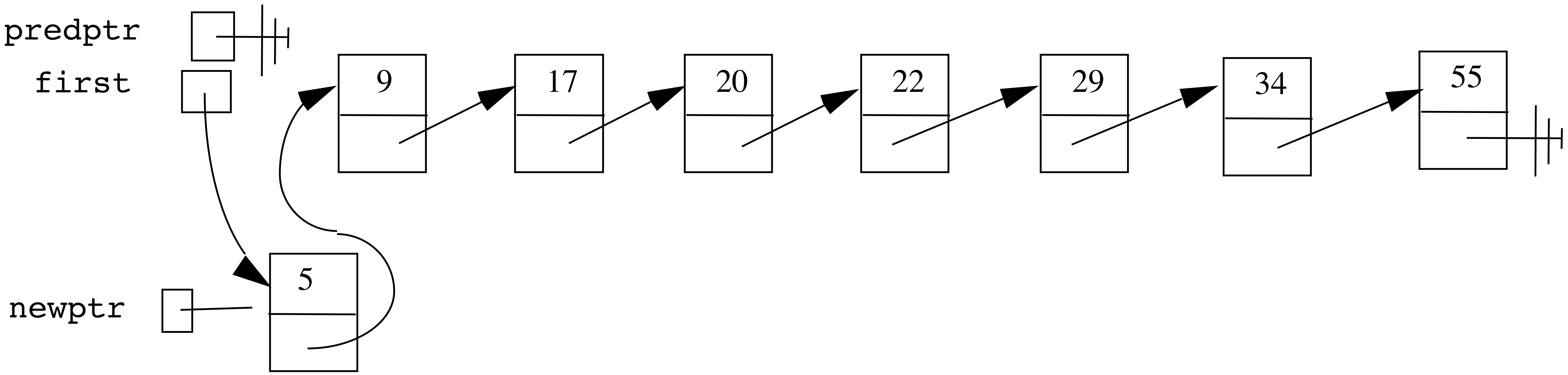


Figure adapted from http://cs.calvin.edu/books/c++/ds/1e/.

Singly Linked Lists

Deletion from Middle

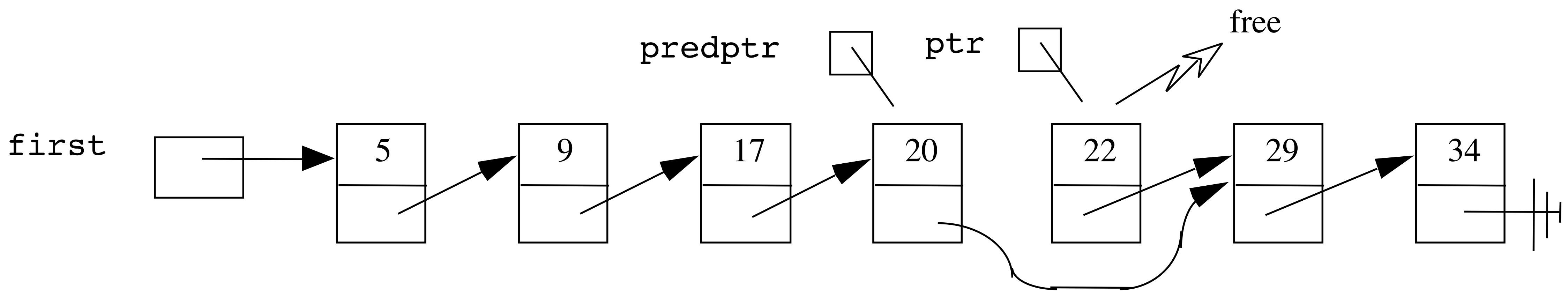


Figure adapted from http://cs.calvin.edu/books/c++/ds/1e/.

Singly Linked Lists

Deletion from Tail

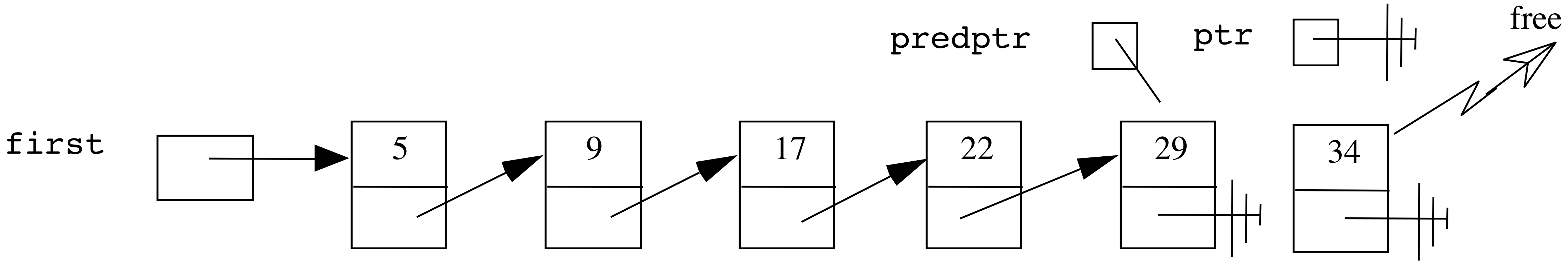


Figure adapted from http://cs.calvin.edu/books/c++/ds/1e/.

Singly Linked Lists

Deletion from Head

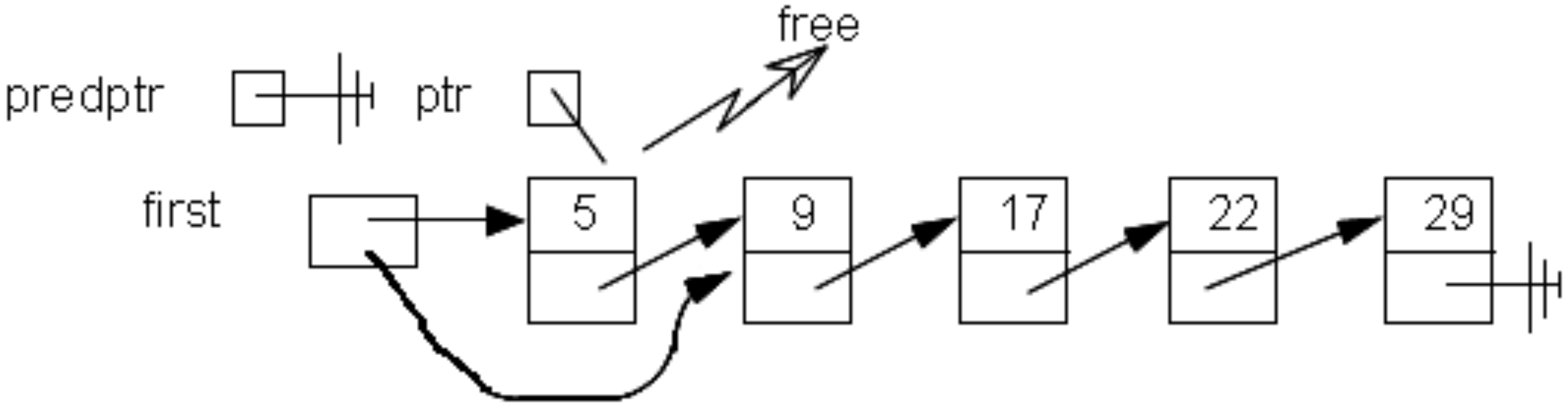


Figure adapted from http://cs.calvin.edu/books/c++/ds/1e/.

Doubly Linked Lists

Representation

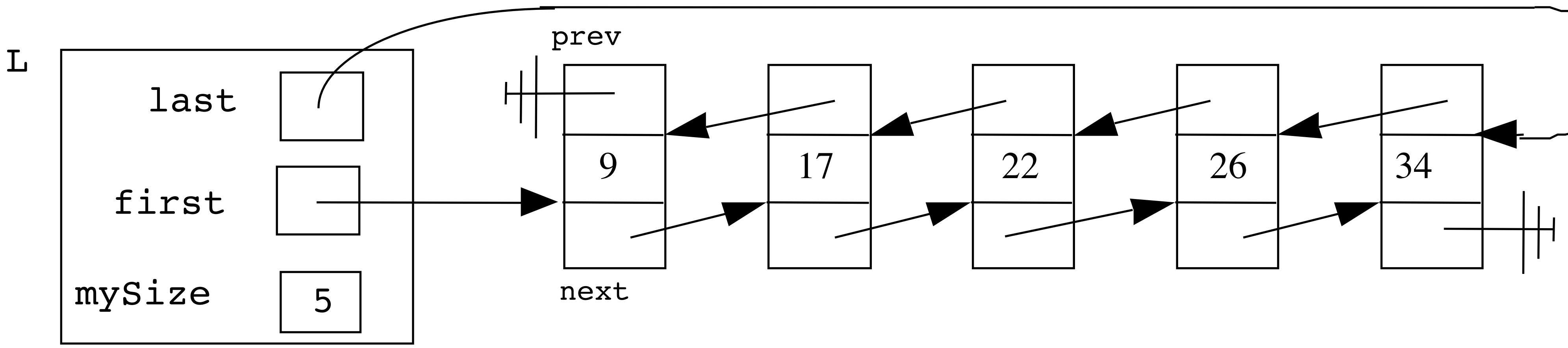


Figure adapted from http://cs.calvin.edu/books/c++/ds/1e/.

Stacks

Queues

Image from http://www.blogcdn.com/www.engadget.com/media/2008/05/iphone_line_1-1.jpg.