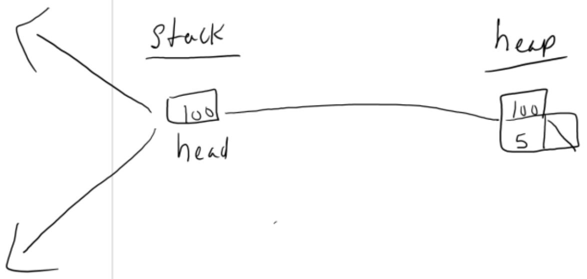
Example 1a (Creating a single node using c++)

```
struct node {
  int data;
  node *next;
};

int main()
{
  node *head;

  head = new node; //Now head points to a node struct
  head->next = NULL;  //The node head points to has its next pointer
                   //set equal to a null pointer
  head->data = 5;  //By using the -> operator, you can modify the node
                   //a pointer (head in this case) points to.
  delete head;
}
```



Example 1b (Creating a single node using c)

```
struct node {
  int data;
  node *next;
};

int main()
{
  node *head;      //This will be the unchanging first node

  head = (node *)malloc(sizeof(struct node));
  head->next = NULL;  //The node head points to has its next pointer
                   //set equal to a null pointer
  head->data = 5;  //By using the -> operator, you can modify the node
                   //a pointer (head in this case) points to.
  free(head);
}
```

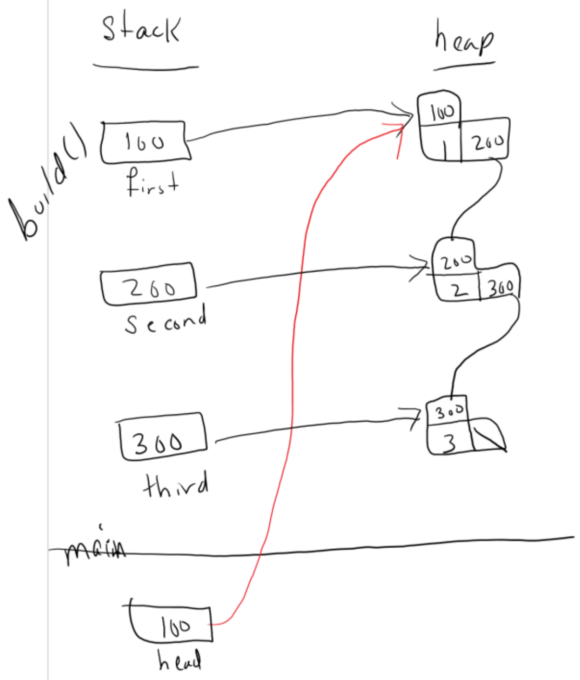
A "head" pointer local to BuildOneTwoThree() keeps the whole list by storing a pointer to the first node.

Each node stores one data element (int in this example).

Each node stores one next pointer.

The next field of the last node is NULL.

```
/*
Build the list {1, 2, 3} in the heap.
Returns the first pointer to the caller.
*/
struct node {
  int data;
  node *next;
};

struct node* build() {
      struct node* first  = NULL;
      struct node* second = NULL;
      struct node* third  = NULL;

      // allocate 3 nodes in the heap
      first  = (node *)malloc(sizeof(struct node));
      second = (node *)malloc(sizeof(struct node));
      third  = (node *)malloc(sizeof(struct node));

      first->data = 1;      // setup first node
      first->next = second; // note: pointer assignment rule

      second->data = 2;     // setup second node
      second->next = third;

      third->data = 3;      // setup third link
      third->next = NULL;

      return first;
}

int main()
{
  node *head=build();
}
```



Stack
heap
build()
100
first
100
1
200
200
Second
200
2
300
300
third
300
3
main
100
head