

7. Pointers, Dynamic Memory

20th September

IIT Kanpur

Agenda

- Pointer to Pointer
 - Dynamic Memory Allocation
 - Pointer to functions
-

Pointer to Pointer

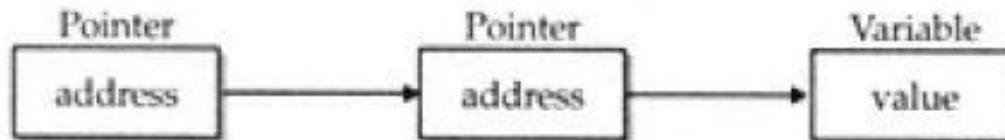
- Declaration
 - Place an additional asterisk

```
double **newbalance;
```

`newbalance` is a pointer to a float pointer.



Single Indirection



Multiple Indirection

Pointer to Pointer contd..

```
#include <stdio.h>

int main() {
    int x, *p, **q;
    x = 10;
    p = &x;
    q = &p;

    printf("%d %d %d\n", x, *p, **q);
    return 0;
}
```

{program: pointers.c}

Dynamic Memory Allocation

- To allocate memory at run time.
- malloc(), calloc()
 - both return a void*
 - you'll need to typecast each time.

```
char *p;  
p = (char *)malloc(1000); /*get 1000 byte space */
```

```
int *i;  
i = (int *)malloc(1000*sizeof(int));
```

Dynamic Memory Allocation contd..

- To free memory
- `free()`
 - `free(ptr)` frees the space allocated to the pointer `ptr`

```
int *i;  
i = (int *)malloc(1000*sizeof(int));  
.  
.  
.  
free(i);
```

Pointers to functions

- A function pointer stores the address of the function.
- Function pointers allow:
 - call the function using a pointer
 - functions to be passed as arguments to other functions

```
return_type (*function_name)(type arg1, type  
arg2...)
```

```
{program: function_pointer.c}
```