

Software Tools

C, Unix (Linux), and tools

If-else

- Typical if-else statement
- The thing to remember that since the else is optional there is an ambiguity:

```
c=0;  
if a<0  
    if b>0  
        c=a+b;  
    else  
        c=a-b;
```

Functions and Program Structure

- We know about functions
- They look like this:
- Declarations
- Definitions

```
type1 funname(type2 var1, ...);  
...  
type1 funname(type2 var1, ...)  
{  
    ...body...  
}
```

Multifile programs

- Any program of descent size comprises several file of C code and headers
 - C code files end on .c
 - Headers end in .h
- They have to share external variables
- We have to observe the cardinal rule of sanity:
 - Never have the same information in two different files
 - Hence the header files!

Program structure

- In the trivial example we are going to do today we have
 - 3 source files
 - One header file
 - 3 object files
 - One executable
- The only way to make order out of chaos is to use makefiles
- Most makefiles for C programming look similar

Makefile

- Our makefile:

```
SRC = main.c stack.c rpolish.c
HDR = rpolish.h
OBJ = main.o stack.o rpolish.o
TRG = rpolish

${TRG}: ${OBJ}
    gcc ${OBJ} -o rpolish

main.o: main.c ${HDR}
    gcc -c main.c -o main.o
stack.o: stack.c ${HDR}
    gcc -c stack.c -o stack.o
rpolish.o: rpolish.c ${HDR}
    gcc -c rpolish.c -o rpolish.o
clean:
    rm -f ${OBJ} ${TRG}
```

Header files

- Our header file:

```
#define MAXOP 100
#define NUMBER 0
#define MAXVAL 100
#define BUFSIZE 100

extern int      getop(char s[]);
extern void     push(double);
extern double   pop(void);
extern int      getch(void);
extern void     ungetch(int c);
```

Header files

- Our header file:

```
#define MAXOP 100
#define NUMBER 0
#define MAXVAL 100
#define BUFSIZE 100

extern int      getop(char s[]);
extern void     push(double);
extern double   pop(void);
extern int      getch(void);
extern void     ungetch(int c);
```


Header files

- Contain
 - Declarations
 - #defines
- May contain #includes to C library headers
- Should not contain definitions
 - Since many programmers mess this up compilers try to guess what it is really meant

Static Variables

- Static variables outside functions
 - Indicate variables private to functions in that file
 - Help avoid name conflicts
- Static variables inside functions
 - Indicate variables private to the function, like regular (automatic) variables
 - Their value survives after the function returns

Problems

- Add the power operator (^) to the abilities of the reverse polish notation calculator.
- Use KRgetline to read one input line at a time. This makes getch-ungetch unnecessary.