

TYPECASTING: (type) expression

ex.

```
float number;  
number = 1/5;    (== 0.0000)  
number = (float) 1/5    (== 0.2000)
```

HEADER FILES:

<stdio.h>
• needed for i/o

<stdlib.h>
<time.h>
• needed for random numbers

<string.h>
• needed for strings

<math.h>
• needed for math op.

GENERATING RANDOM INTEGERS:

- need both `stdlib` and `time` header files always
- need `srand(time(NULL));` within `int main()` to set the seed

→ general formula: $\text{rand()} \% (\text{MAX} - \text{MIN} + 1) + \text{MIN}$

ex. generate in range from 0 to N
 $\text{rand()} \% (N + 1)$

ex. generate in range from 1 to N
 $\text{rand()} \% N + 1$

ex. generate in range from K to N
 $\text{rand()} \% (N - K + 1) + K$

→ to generate a random float, typecast with `(float)`
ex. $(\text{float}) \text{rand()} / N$

SELECTION STATEMENTS

```
• if (condition expression)  
{  
    general statement;  
    other statement;  
}
```

```
else  
    different statement;
```

• switch example:

```
switch (variable)  
{  
    case 0:  
    case 1:  
        printf("For case 0 or 1\n");  
        break;  
    case 2:  
        printf("For case 2\n");  
        break;  
    default:  
        printf("No case\n");  
}
```

LOOPS!!!

- for loop:

```
for (i = 0; i <= N; i++)  
    printf("#i!\n")
```

- nested loop:

```
for (i = 0; i <= N; i++)  
{  
    //action in outer loop, including inner  
    for (j = 0; j <= K; j++)  
    {  
        //action in inner loop  
    }  
}
```

- while loop: action may be skipped entirely

```
while (condition expression)  
{  
    action1;  
    action2;  
}
```

- do while loop: action on top, therefore always executed at least once!

```
do  
{  
    action1;  
    action2;  
}  
while (condition expression);
```

FILE I/O

FILE *ifp,
*ofp;

functions: fopen to open, fclose to close,
fscanf to read from file, fprintf to
write or append

ex. of opening/closing

```
ifp = fopen("oldfile.dat", "r"); //reading
ofp = fopen("newfile.dat", "w"); //writing
ofp = fopen("newfile.dat", "a"); //appending
fclose(ifp)
fclose(ofp)
```

ex. reading from file

```
fscanf(ifp, "%f", &inputnum);
```

ex. writing/appending to file

```
fscanf(ofp, "%f", inputnum); //note no & on inputnum
```

ex. with while loop reading/writing

```
while (fscanf(ifp, "%f", &inputnum) != EOF)
    fprintf(ofp, "%f\n", ceil(inputnum));
```

STRINGS ... don't forget string.h !!!

- strcpy(strvar, str);

ex. strcpy(mystr1, "Leah");
printf("In the variable 'mystr1' is %.s now!\n", mystr1);

- strlen(str); returns string length w/out end of string sentinel

ex. strlen(mystr1) ... == 4

- strcat(str1, str2); concatenates str2 to end of str1

ex. strcpy(str1, "base");
strcpy(str2, "ball");
strcat(str1, str2); ... str1 == "baseball"

- strcmp(str1, str2); just compares the two strings, returning
0 if they're the same or +/- values