CS201 Discussion Section 1 Notes

Week of Feb 10, 2001

Structures

There are three basic types of structures in C:

- Sequential structures
- Selection structures
- Repetition structures

Sequential structures

Sequential structures are simply a list of expressions that will be evaluated in order, for example:

```
printf("Hello World")
scanf("%d", &i);
printf("i = %d", i);
```

Selection structures

if example

```
if (i == 1) {
    printf("one\n");
} else if (i == 2) {
    printf("two\n");
} else if (i == 3) {
    printf("three\n") {
} else {
    print("num\n");
}
```

switch example

Doing the same thing but with a switch:

```
switch (i)
{
    case 1:
        printf("one\n");
        break;
    case 2:
        printf("two\n");
        break;
    case 3:
        printf("three\n");
        break;
    default:
        printf("num\n");
        break;
}
```

the purpose of break

If i is 2 then the program should print “two”. However if the switch statement was:

```
switch (i)
{
    case 1:
        printf("one\n");
    case 2:
        printf("two\n");
    case 3:
        printf("three\n");
    default:
        printf("num\n");
}
```

then the program would of printed:

```
two
three
num
```

because with out the break statements C will continue to execute the remaining lines of code until it finds a break statement. So....

don’t forget the break statements

Repetition structures

for loops

```
int i;
for (i = 0; i != 3; i++) {
    printf("%d\n", i);
}
```

do while loops

```
int i;
scanf("%d", i);
do {
    printf("%d\n", i);
    i++;
} while (i < 3);
```

If the user enters 0 then this do/while loop is the same as the for loop. If the user enters 5 then the program will print “5” and quit.

while loops

```
int i;
scanf("%d", i);
while (i < 3) {
    printf("%d\n", i);
    i++;
}
```

This is *not* the same code as the previous do/while loop because if a user enters 5 then nothing will be printed as the condition is checked *before* the body is executed in a while loop but *after* in a do/while loop.

Boolean Variables

C does not have a true boolean variable. In C any nonzero integer is considered true and a 0 is considered false. For example

```
int i,x,y;
x = 10;
y = 20;
i = x < y;
if (i) {
    /* this code should execute since i is non zero */
}
```

```
    i = x > y;
    if (i) {
        /* this code should not execute since i is zero */
    }
```

What’s Wrong with this Code

Example 1

```
    #include <stdio.h>
    void main()
    {
        float i;

        for (i = 0; i != 12; i++);
        {
            if (i % 3 = 0)
            {
                printf "%f" i;
            }
        }
    }
```

Example 2

```
    include <stdio.h>
    int main()
    {
        int num;

        printf("Please enter in a number: ");

        scanf("%d", num);

        if (num != 2 || num != 3)
        {
            printf("Your number does not equal 2 or 3\n");
        }

        return 0;
    }
```