CSC 215
Control Flow

Dr. Achraf El Allali

If-else

```
if ( grade== ‘A’)
{
        printf(“Great!”);
}
else
{
        printf(“Try again”);
}
```

- Braces are optional for single statements

else-if

```
if ( grade== ‘A’)
        printf(“Great!”);
else if (frade==’B’)
        printf(“Good!”);
else if (grade==’C’)
        printf(“OK!”);
else
        printf(“Try again”);
```

Nested if

```
#include <stdio.h>
int main(){
        int v1, v2;
        printf(“\nEntertwo values: ”); /* Enter two numbers */
        scanf(“%i %d”, &v1, &v2);
        if (v1 > v2){ printf(“\n%1d is > %1d\n”, v1, v2);}
        else {
        if (v1 < v2){ printf(“\n%1d is < %1d\n”, v1, v2);}
        else {printf(“\n%1d is = %1d\n”, v1, v2);}
         }
return 0;
}
```

switch

```
switch (grade) {
        case ‘A’:
            printf(“Great!”);
            break;
        case ‘B’:
            printf(“Good!”);
            break;
        case ‘C’:
            printf(“Great!”);
            break;
        default:
             printf(“Try again”);
             }
```

Puzzle

```
int  m=1;
switch(m)
{
        case 0:
        printf("Zero\n");
        case 1:
        printf("One\n");
        m++;
        case 2:
        printf ("Two\n");
        case 3:
        printf ("Three\n");
}
printf ("m = %d\n",m);
```

While statement

```
int a[10] = {1,2,3,4,5,6,7,8,9,10};
int i = 0, sum = 0;
while (i < 10)
{
    sum = sum + a[i];
    i++;
}
printf("Average = %f", sum/10);
```

for

```
int a[10] = {1,2,3,4,5,6,7,8,9,10};
int i, sum = 0;
for (i = 0; i < 10; i++)
{
    sum = sum + a[i];
}
printf(“Sum = %d”, sum);
```

do-while

```
do
{
    scanf(“%c”, &ch);
}
while (ch != 'z');
```

Loop interruption

- C provides two statements for implementing loop interruptions:
  - break statement
  - continue statement
- When a break statement is encountered within a loop body, the execution of the loop body is interrupted, and the program control transfers to the exit point of the loop
- Within a nested loop, break statement results in interruption of the innermost loop whose body contains the break statement

break

```
for (i = 0; i < n; i++)
{
    if (a[i] < 0)
        break;
    sum = sum + a[i];
}
```

Loop interruption

- The continue statement does not terminate the loop; it only interrupts a particular iteration
- When a continue statement is encountered within a loop body of a while or do-while loop, all the remaining statements in the loop body are skipped and the loop continuation condition is evaluated next
- Within the for loop, any statements in the loop body are skipped, and the re-initialization expression (the third one) is evaluated next
- Then the execution of the repetition continues as normal

continue

```
for (i = 0; i < n; i++)
{
    if (a[i] < 0)
        continue;  /*Skip negative no.s */
    sum = sum + a[i];
}
```