Lecture 4

printf() and scanf()

```
#include <stdio.h>
int a;
double b;
scanf("%d", &a);
scanf("%lf", &b);

printf("\t␣a=%d␣\t␣b=%lf␣\n", a, b);
```

- Branching is deciding what actions to take
- The program chooses to follow one branch or another
- *if()*
- *? :* operator
- *switch()*

if()

If today is Monday I will study C programming

- Based on a concept of *TRUE* or *FALSE*
- *TRUE* is a statement that evaluates to a nonzero value
- *FALSE* evaluates to zero
- Use of relational operators:
  - $>$ greater than - $5 > 4$ is TRUE
  - $<$ less than - $4 < 5$ is TRUE
  - $>=$ greater than or equal - $4 >= 4$ is TRUE
  - $<=$ less than or equal - $3 <= 4$ is TRUE
  - $==$ equal to - $5 == 5$ is TRUE
  - $!=$ not equal to - $5 != 4$ is TRUE
- exapmles ...

Do not use = to test equality, use == !!!

AND and OR

&& and ||

- Used for more complex logical statements
- && - logical AND
- || - logical OR

Basic if syntax

```
if (statement that evaluates to TRUE or FALSE)
  instruction

if (statement that evaluates to TRUE or FALSE)
{
  multi
  line
  instruction
}
```

examples ...

What else?

```
if (statment that evaluates to TRUE or FALSE)
  instruction
else
  another set of instructions

if (statment that evaluates to TRUE or FALSE)
{
  multi
  line
  instruction
}
else
  another set of instructions - could be multiline
```

examples ...

else if()

```
if (statment that evaluates to TRUE or FALSE)
  instruction
else if()
  another set of instructions
else if()
 ...

if (statment that evaluates to TRUE or FALSE)
{
  multi
  line
  instruction
}
else if()
  another set of instructions - could be multiline
```

examples ...

Inline if

?:

- It is like an *if else*
- Might be used within expressions
- The only ternary operator in C

```
if condition is true ? then X return value : otherwise Y value;

int a=5;
int b=1, c=2;
int d = b > c ? a + b : a + c;
```

examples ...

switch()

- Much like nested if else
- Might be more efficient

```
switch( expression )
{
  case expr1:
    instructions;
    break;
  case expr2:
    instructions;
    break;
  default:
    instructions;
}
```

- key word *break*
- key word *default*
- examples...