If

Conditional Processing

Simple If statement syntax

if (<condition>) <then_statement;>

- <condition> : Any expression whose results is true or false
- <then_statement;> : Any statement or block of statements
- The <then_statement> is executed only when the <condition> is true
- Warning: No “then” keyword!

```
if (x!=0) x=113/x;
```

Example If statement

```
if (temp > 35) {
        shut_down();
        printf("Maximum temperature exceeded… shut down\n");
        return ERROR;
}
process_more();
```

If/Then/Else syntax

if (<condition>) <then_statement;> else <else_statement;>

- <condition> : Any expression whose results is true or false
- <then_statement;> : Any statement or block of statements
- <else_statement;> : Any statement or block of statements
- The <then_statement> is executed only when the <condition> is true
- The <else_statement> is executed only when the <condition> is false

if (x!=0) x=113/x; else x=-1;

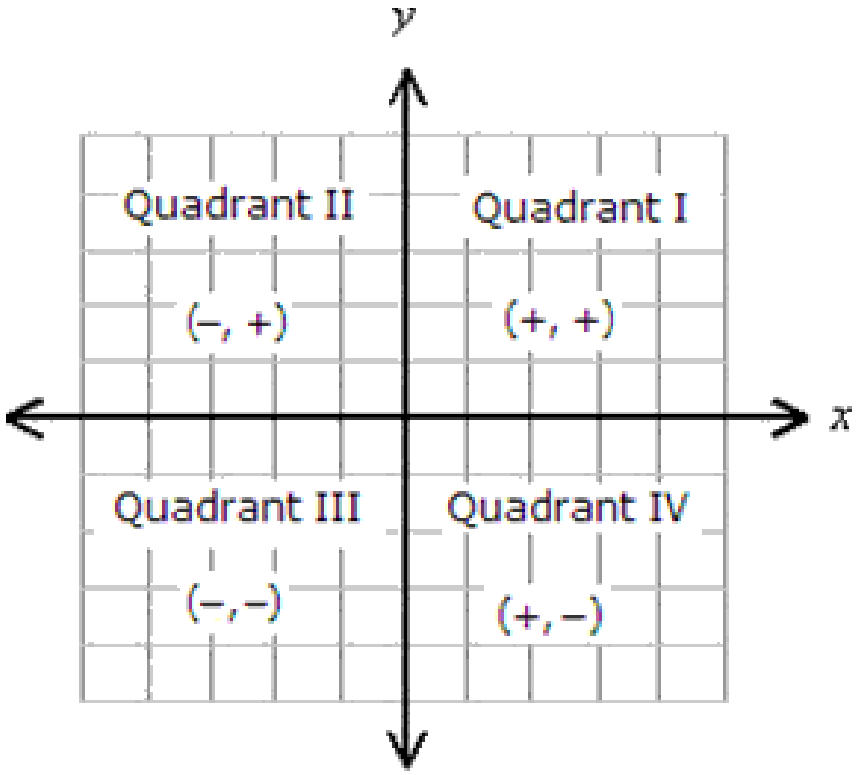
If/Then/Else Example

```
float guess=1;
if (guess*guess < num) {
        guess = guess + (num – guess*guess)/2;
} else {
        guess = guess – (guess*guess – num)/2;
}
```

Nested If/Then/Else

- It's perfectly legal for a then_statement or else_statement to be an if/then/else statement

```
if (x>0) {
        if (y>0) printf("Quadrant I");
        else printf("Quadrant IV");
} else {
        if (y>0) printf("Quadrant II");
        else printf("Quadrant III");
}
```



Alternative… Compound Logic

```
if ((x>0)&&(y>0)) printf (“Quadrant I”);
if ((x>0)&&(y<0)) printf(“Quadrant IV”);
if ((x<0)&&(y>0)) printf(“Quadrant II”);
if ((x<0)&&(y<0)) printf(“Quadrant III”);
```

Alternative Else/If Construct

```
if      ((x>0)&&(y>0)) printf (“Quadrant I”);
else if ((x>0)&&(y<0)) printf(“Quadrant IV”);
else if ((x<0)&&(y>0)) printf(“Quadrant II”);
else if ((x<0)&&(y<0)) printf(“Quadrant III”);
else                   printf(“Origin”);
```

Nested If/Then/Else Ambiguity

```
if (x>0) if (y>0) printf (“x>0, y>0\n”);
else printf (“Is x negative, or is y negative????\n”);

if (x>0) { if (y>0) printf (“x>0, y>0\n”);
             else printf(“y<0\n”);
         }

if (x>0) { if (y>0) printf (“x>0, y>0\n”); }
else printf(“x<0\n”);
```

Using “return” instead of “else”

```
if (x<y) {
        printf(“x is less”);
} else {
        printf(“y is less”);
        y = x +3;
        printf(“Now x is less”);
}
return;
```

```
if (x<y) {
        printf(“x is less”);
        return;
}
printf(“y is less”);
y=x+3;
printf(“Now x is less”);
return;
```

Don't Forget "?"

```
if (doMore) {
        x=y*3;
} else {
        x=x-1;
}
```

```
x=doMore?(y*3):(x-1);
```

Another else/if construct

```
if      (op==‘+’) ans=a+b;
else if (op==‘-’)  ans=a-b;
else if (op==‘*’)  ans = a*b;
else if (op==‘/’)  ans = a/b
else {
        printf “Unrecognized operator: %c\n”,op);
        ans=0;
}
```

The "switch" statement

```
switch(expression) {
        case (v1) :
                v1_statement;
                v1_statement;
                break;
        case (v2) :
                v2_statement;
                v2_statement;
                break;
        …
        default:
                def_statement;
                def_statement;
}
```

```
if (expression==v1) {
        v1_statement;
        v1_statement;

} else if (expression==v2) {
        v2_statement;
        v2_statement;
}
…
} else {
        def_statement;
        def_statement;
}
```

Example Switch Statement


```
switch(op) {
        case('+'): ans=a+b; break;
        case('-'): ans=a-b; break;
        case('*'): ans=a*b; break;
        case('/'): ans=a/b; break;
        default: printf("Unrecognized operator: %c\n",op);
                ans=0;
}
```

Resources

- Programming in C, Chapter 5
- Wikipedia: Conditional (computer programming) (https://en.wikipedia.org/wiki/Conditional_(computer_programming))