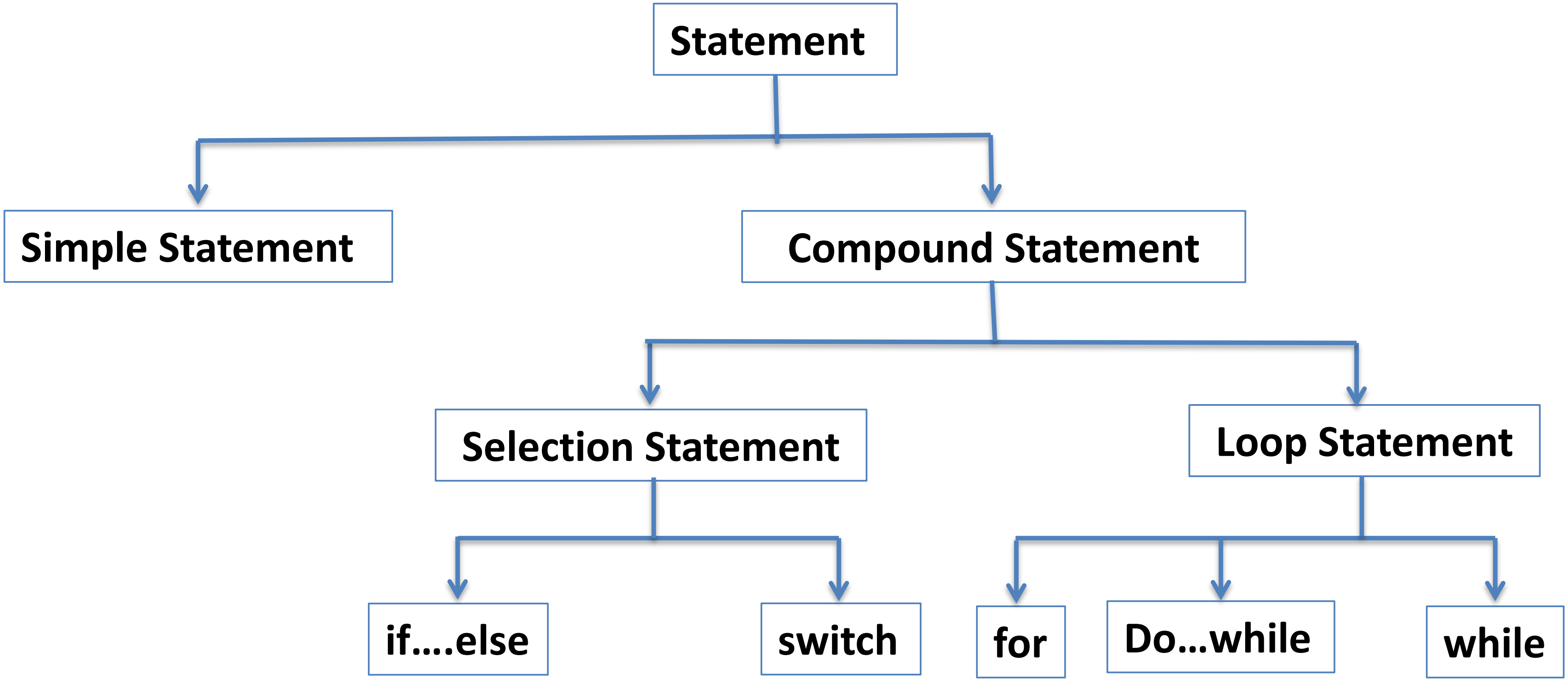
Control Structure

Statement

- **Statement:** A statement is a command that instructs computer to perform a specific action.



Statement

Simple Statement

Compound Statement

Selection Statement

Loop Statement

if….else

switch

for

Do…while

while

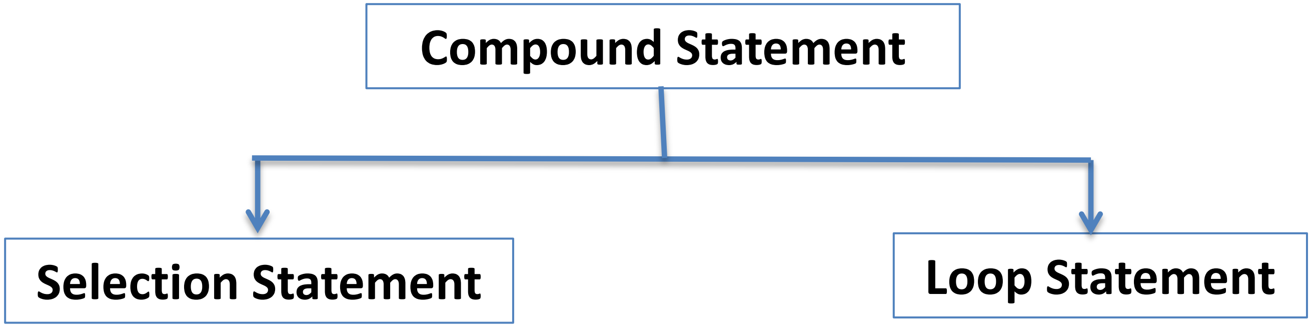
Simple Statement

Simple statement is of different types:

Simple Statement	Examples
Expression	++count; break;
Equation	c = (a+b)/2*d; a+=2;
Function	printf(“Enter your name: “);
Variable declaration	int a, b, c = 2;
Function prototype	float add(float x, float y);

Compound Statement

- **Compound Statement:** A compound statement may contain more than one statements.

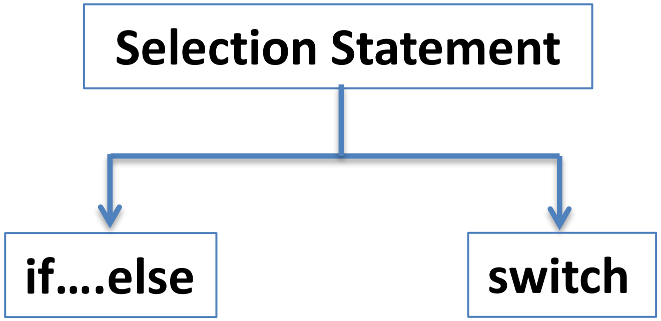

Compound Statement

Selection Statement

Loop Statement

Selection Statement

- **Selection Statement:** In a selection statement, a single or block of statements within {} will be either executed or not based on the value of some condition.



Selection Statement

if….else

switch

if…else…. Statement

- **Structure of if-else statement:**

```
if (cond.)
        ……………….;
```

```
if (cond.){
        ……………….;
}
```

```
if (cond.){
        ……………….;
        ……………….;
}
```

```
if (cond.)
        ……………….;
else
        ……………….;
```

```
if (cond.)
        ……………….;
else{
        ……………….;
        ……………….;
}
```

- **Nested-if Statement:** If an "if" statement is embedded inside another "if" statement, then it is called "nested if" statement.

if…else…. Statement

Exercise: Write a C program to solve the quadratic equation, $ax^2 + bx + c = 0$

Solution: Mathematical Analysis,

$$x = \frac{-b \pm \sqrt{b^2 - 4ca}}{2a}$$

Assume, $d = b^2 - 4ca$

$$x1 = \frac{-b + \sqrt{b^2 - 4ca}}{2a}$$

$$x1 = \frac{-b + \sqrt{d}}{2a}$$

$$x2 = \frac{-b - \sqrt{b^2 - 4ca}}{2a}$$

$$x2 = \frac{-b - \sqrt{d}}{2a}$$

Program Planning:

Variables: a, b, c, d, x1, x2
All are float/double type.

Unsolvable Conditions:
(i) if $a = 0$
(ii) if $d < 0$

Algorithm:

1. Read a, b, c
2. If $a = 0$, exit from program
3. Calculate d
4. If $d < 0$, exit from program
5. Calculate x1 and x2
6. Print x1, x2

if…else…. Statement

```
#include <stdio.h>
#include <math.h>

main(){
    float a, b, c, d, x1, x2;

    printf(“Enter value of a: “);
    scanf(“%f”, &a);
    printf(“Enter value of b: “);
    scanf(“%f”, &b);
    printf(“Enter value of c: “);
    scanf(“%f”, &c);
```

```
    if (a){
        d = b * b – 4 * c * a;
        if (d >= 0){
              x1 = (-b + sqrt(d))/(2*a);
              x2 = (-b - sqrt(d))/(2*a);
              printf(“X1 = %.2f     X2 = %.2f”, x1, x2);
        }
        else
             printf(“Error! Roots are imaginary.”);
    }
    else
        printf(“Error! Non-quadratic Equation.”);
    return 0;
}
```

switch Statement

❖ Structure of switch statement:

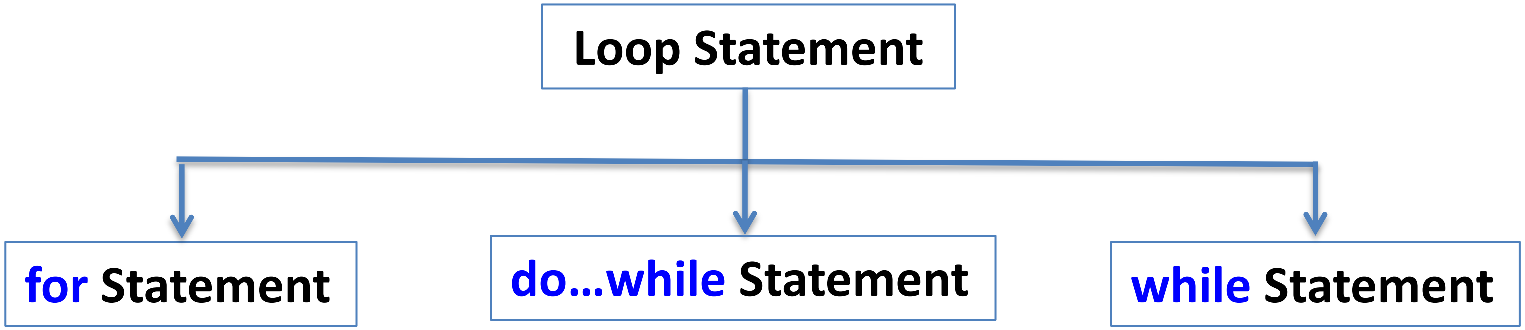
```
switch (expression){
case …..:  …………….;
                 …………….;
                 break;
case ……: ……………..;
                 ……………..;
                 break;
default: ………………..;
                 ………………..;
}
```

Exercise:

```
choice = getchar();

switch( choice ){
case ‘r’:
case ‘R’: printf(“RED”);
                 break;
case ‘w’:
case ‘W’: printf(“WHITE”);
                 break;
case ‘b’:
case ‘B’: printf(“BLUE”);
                 break;
default: printf(“NONE”);
}
```

Loop Statement



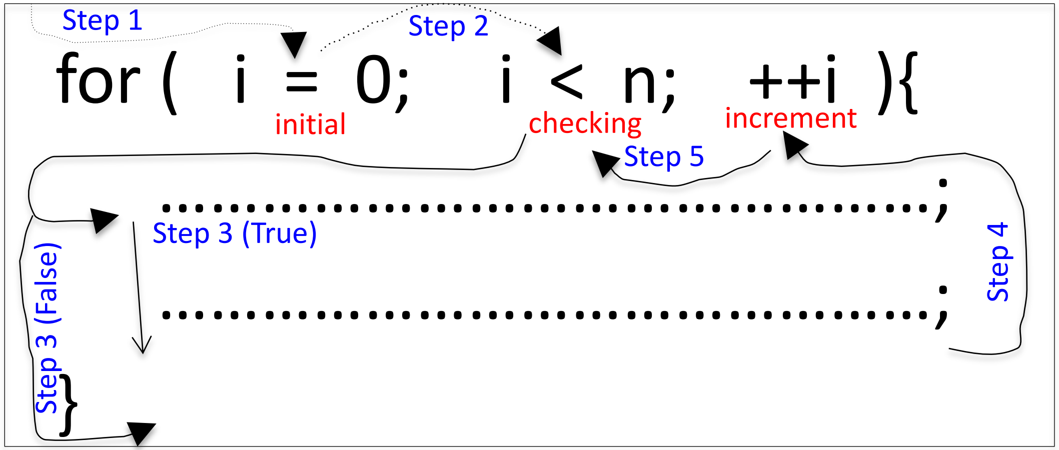
Loop Statement

for Statement

do…while Statement

while Statement

❖ Structure of for statement:



```
for (  i = 0;   i < n;   ++i  ){
        ..........................................;

        ..........................................;
}
```

Flow of Cursor:

- ✓ **Step 1**: Initial part will be executed only once at the starting of the loop;
- ✓ **Step 2**: From initial part, cursor will move to checking part;
- ✓ **Step 3**: If checking part is true, cursor will move to body of the loop. If checking part is false, cursor move out of the loop;
- ✓ **Step 4**: After completion of the loop, cursor will move to increment part;
- ✓ **Step 5**: After completion of increment part, cursor will move to checking part; and
- ✓ **Step 6**: move to step 3 for further movement.

for Statement

Exercise: Write a C program to find out the sum of n integer numbers. The value of n and all other numbers will be taken as inputs.

Solution:

```
#include <stdio.h>

main(){
   int i, n, x, sum = 0;

   printf(“Enter value of n: “);
   scanf(“%d”, &n);
   for ( i = 0; i < n; ++i){
          printf(“Enter number %d: “, i+1);
          scanf(“%d”, &x);
          sum += x;
   }
   printf(“The sum is %d.”, sum);
}
```

while Statement

❖ Structure of while statement:

```
while (cond){
        ...........................;
        ...........................;
}
```

Execution:

✓ Condition is checked; If the condition is true, execute the statements inside while and then come back to the condition again. If the condition is false, then go outside of the loop.

Exercise: Convert the following for loop into equivalent while loop.

```
for ( i = 0; i < n; ++i){
      printf(“Enter number %d: “, i+1);
      scanf(“%d”, &x);
      sum += x;
 }
```

Solution:

```
i = 0;
while (i < n){
      printf(“Enter number %d: “, i+1);
      scanf(“%d”, &x);
      sum += x;
      ++i;
 }
```

while Statement

Exercise: Write down a C program to find sum of all inputted integer numbers before giving 0 as an input. (use while loop)

Solution:

```
#include <stdio.h>

main(){
    int i = 0, x, sum = 0;

    printf(“Enter number %d: “, i+1);
    scanf(“%d”, &x);
    while ( x ){
           sum += x;
           ++i;
           printf(“Enter number %d: “, i+1);
           scanf(“%d”, &x);
     }
     printf(“The sum is %d.”, sum);
}
```

do...while Statement

❖ Structure of while statement:

```
do {
        ………………………….;
        ………………………….;
} while( cond );
```

Execution:

✓ At first the statements inside the loop will be executed and then condition is checked. If the condition is true comeback to do label; otherwise go out of loop.

Exercise: Write down a C program to find sum of all inputted integer numbers before giving 0 as an input. (use do … while loop)

Solution:

```
#include <stdio.h>

main(){
    int i = 0, x, sum = 0;

   do{
        printf(“Enter number %d: “, i+1);
        scanf(“%d”, &x);
        if (x){
            sum += x;
            ++i;
        }
   } while (x);
   printf(“The sum is %d.”, sum);
}
```

Assignments

Assignment 1: Write a C program to find n!.

Assignment 2: Write a C program to solve the following equation:

$$x^5 + 3x^2 – 10 = 0.$$

Assignment 3: Write a C program to calculate the weighted average of a list of n numbers.

$$x_{avg} = x_1f_1 + x_2f_2 + \ldots\ldots\ldots\ldots\ldots + x_nf_n$$

Assignment 4: Write a C program to find out whether an inputted number prime or not.

Assignment 5: Write a C program to create Pascal’s triangle based on inputted value.