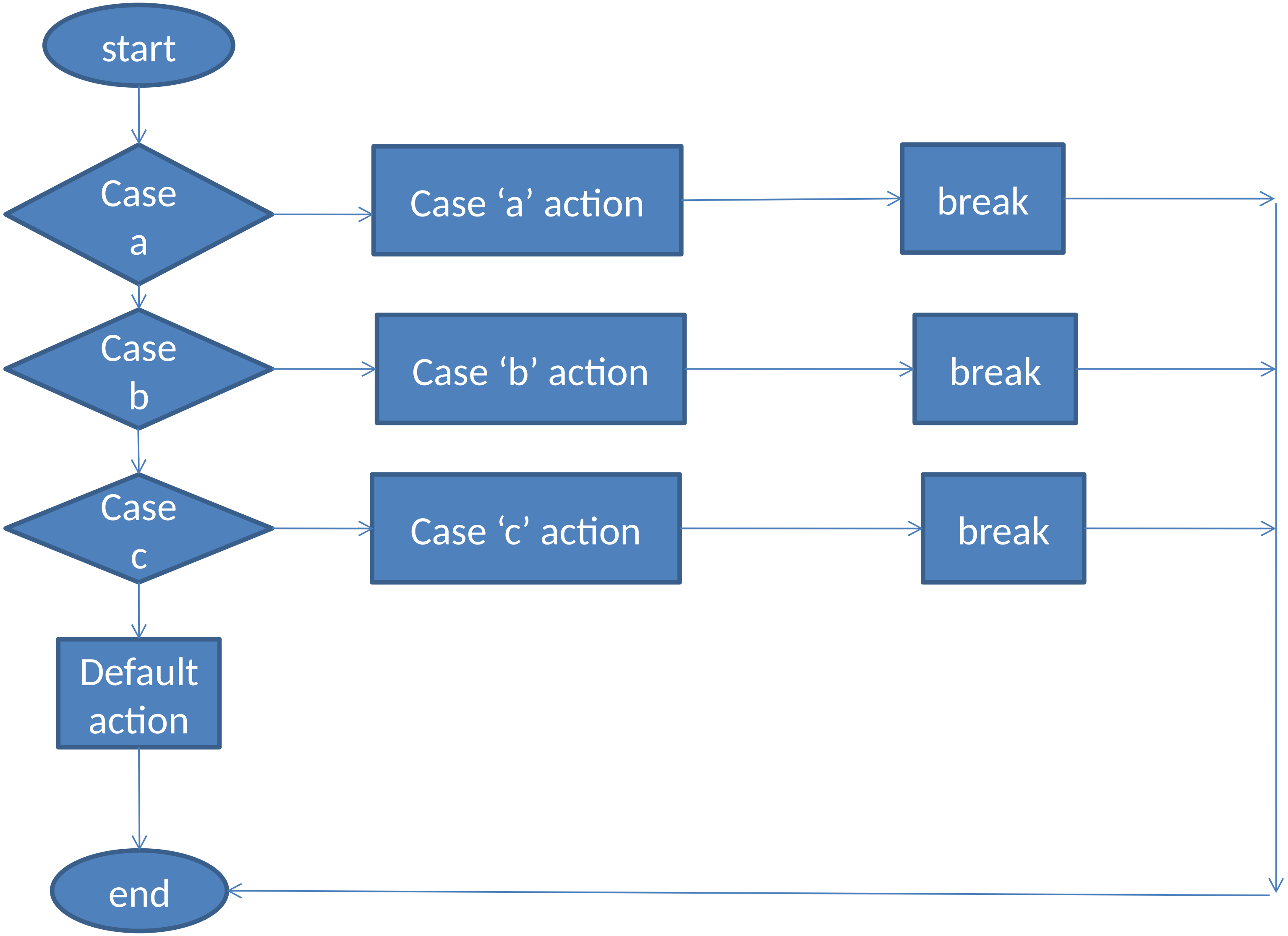
The switch multiple -selection structure.

The Switch Structure

- If is a single- selection structure.
- If/else is a double selection structure.
- Switch is a Multiple- selection structure.

The switch structure consists of a series of case labels, and an optional default case.

Flow chart for multiple –selection structure



Syntax

```
switch (var)
{
case label:  statements; break;
case label:  statements; break;
default: statements; break;
}
```

Example

```
int value=0;
void setup()
{
 Serial.begin(9600);
}
void loop()
{
  value=value+1;
  switch (value)
  {
    case 1: Serial.println( "Monday");delay (1000); break;
    case 2: Serial.println ("Tuesday");delay(1000);  break;
    case 3: Serial.println ("Wednesday"); delay(1000); break;
    case 4: Serial.println ("Thursday"); delay (1000);break;
 case 5: Serial.println ("Friday"); delay (1000); break;
  case 6: Serial.println ("Saterday");delay (1000);break;
  case 7: Serial.println ("Sunday"); delay(1000); break;

default:
    Serial.println(" There are no other day in the week");
    delay (1000);
    break;

 }
}
```

Repetition statements

The followings builds structure within which the repetition is based on some logical and/or arithmetic operations.

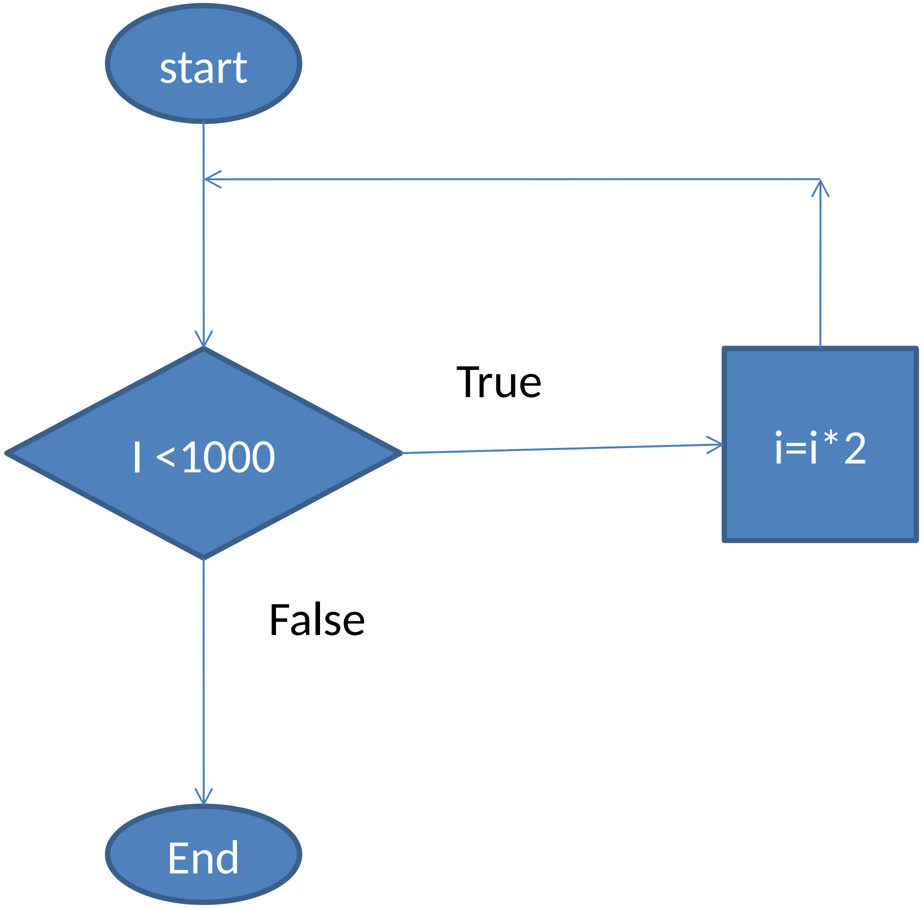
- While structure
- Do/ while structure
- For structure
- Counter control repetition

While Structure

The pseudo-code statement:

```
While there are more items on my shopping list
Purchase next item and cross it off my list
```

While - loop flow chart



syntax

Syntax

```
while(expression)
{
    statement(s)
}
```

Parameters

expression - a (Boolean) “C” statement that evaluates to true or false

Example

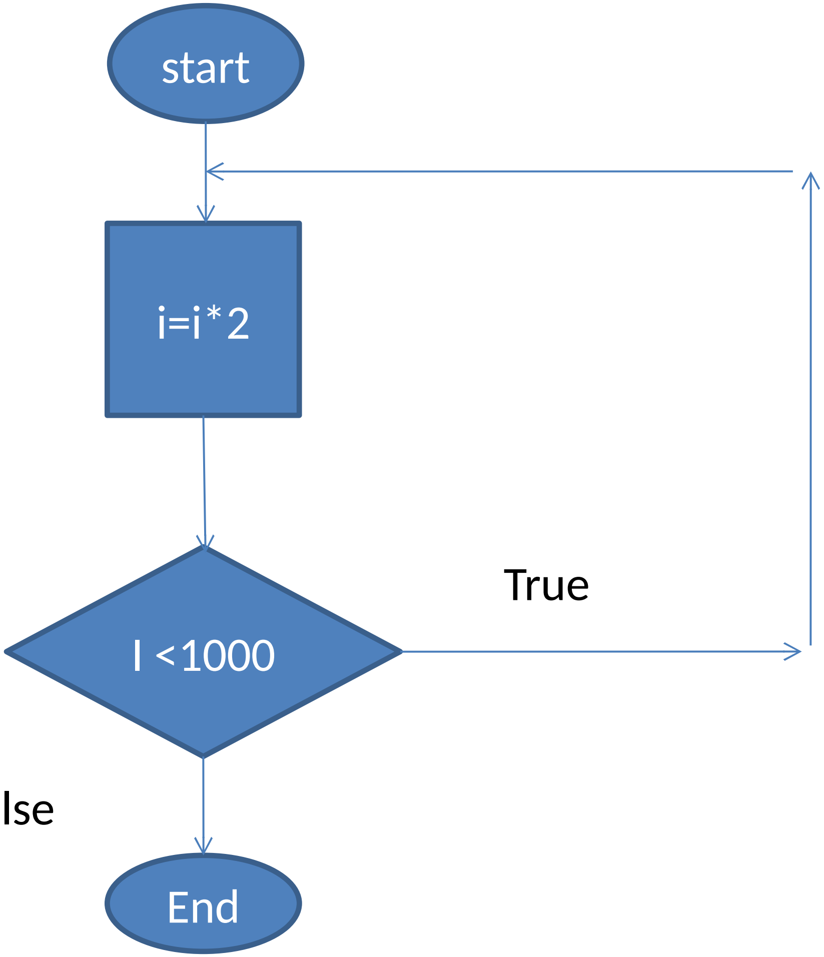
```
int Count = 0;
while(Count < 200)
 {
    // do something repetitive 200 times
   Serial.println( “I am learning C”);
    Count++;
  }
```

Do/ while structure

The while structure, tests the condition at the beginning of the loop to decided, if repetition is to be done.

The do-while structure, the operation in the main loop is done at least once, before the condition is tested at the end of the loop.

The Flow chart for do-while



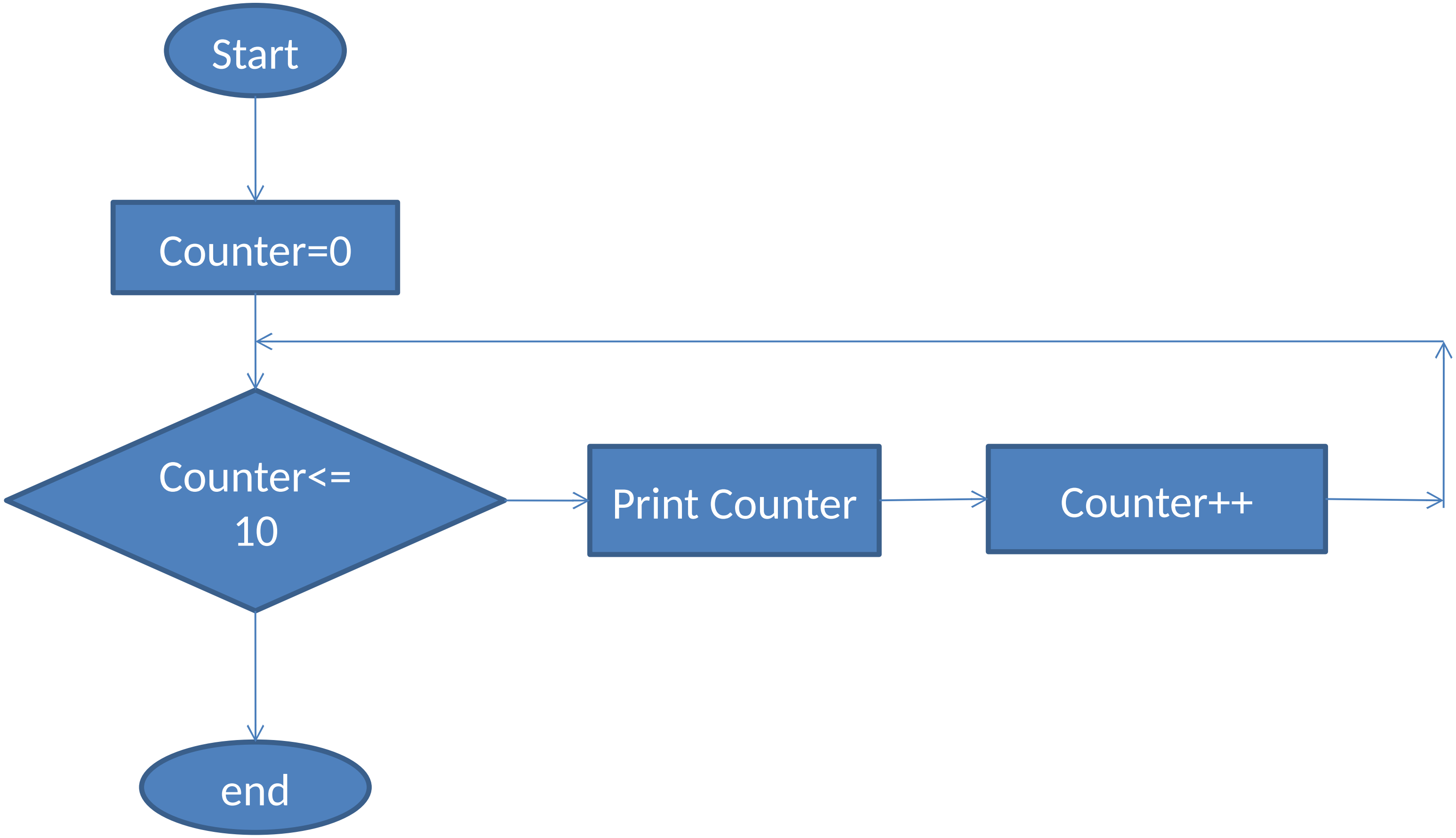
Example

```
int J = 0;
do
 { // do something repetitive 200 times
     Serial.println( “I am learning C”);
       J++;
  }
while(J< 200);
```

Counter Control Repetition

- Needs a control variable
- Initial value of the control variable
- The increment
- The condition that tests the final value of the control variable

Flow chart



Example /for statement

```
for (counter=0; counter<=10; counter++)
    {
      Serial.println(“Counter=“ counter);
    }
```

Example While statement

```
while(counter< 100)
    {
      Serial.println(“Counter=“ counter);
    counter=counter+1;
    }
```

Do /while example

```
do
  {
   Serial.println(“Counter=“ counter);
    counter=counter+1;
    }
while (counter =< 100);
```

Structured programming summary

- **Sequence has no branching**
- **Selection**

  if, if-else , switch
- **Repetition**

  while, do, do/ while

The break and continue statements

- They are used to change the flow of the program.
- Causes immediate exit from the structure.
- Execution continues with the first statement after the structure.
- Escapes from a loop

Break Example

```
int loop ()
 {
   int x;
   for (x=1; x<=10 x++)
      {
         if(x==5)
       break;  /*break loop only if x==5 */
         Serial.print(“ X=“);
       Serial.println(x);
       }
     Serial.print(“Broke out of the loop at x=“);
     Serial.println (x);
  }
```

Continue Example

```
int loop ()
 {
  int x;
  for (x=1; x<=10; x++)
    {
      if (X==5)
        continue;  /*skip the remaining code in loop only if x== 5 */
         Serial.print(“ X=“);
  Serial.println(x);
     }
   Serial.println(“ Used continue to skip printing the value of 5);
  }
```