Programming Principles

Unit 3

Loops 1

Looping (iteration)

Repeating stuff

Iteration

- **Iteration is the second fundamental concept in program design**
- **The first one is Sequence**
- **Iteration means repetition or looping**
- **Used whenever a program or part of a program needs to be repeated**

This week we shall concentrate on loops that repeat a known number of times.
To do this we shall use:

- the while() loop
- the for() loop

while(…) loop

Used to repeat
a block of code

Syntax (grammar rules)

The **while(…)** loop keeps going as long as a **condition** is **true**

```
extern void object::whileLoop()
{
        //   the beginning of program goes here

           while ( condition )
           {
              // block of code
              // to be repeated
              // goes here …
              //  block keeps repeating while condition is true
           }

        //   the rest of the program continues when the loop finishes
}
```

a **condition** is something that is either **true** or **false**

An Example

Display a message 10 times

```
extern void object:: Hello()
{
        int count;      // set up a counter for the loop
        count = 0;      // initialise counter to zero
        while ( count < 10 )
        {
            message ("Hello Friends");
            wait (0.5);
            count ++;
        }

        message ("Loop Now Finished");
}
```

while count **is less than** 10

add 1 to count

More conditions

dist <= 20

count < 10

count > 5

i < 30

angle >= 90

fname == "Otto"

counter != 0

altitude == 20

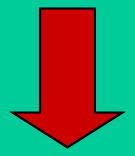
Relational Operators

<	…. Less than
>	…. Greater than
<=	…. Less than or equal to
>=	…. Greater or equal to
==	…. Equal to
!=	…. Not equal to

All of these are **boolean** items (either **true** or **false**)

increment (++) & decrement(--) operators

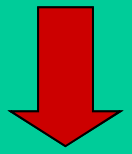
```
count ++;
```



is equivalent to

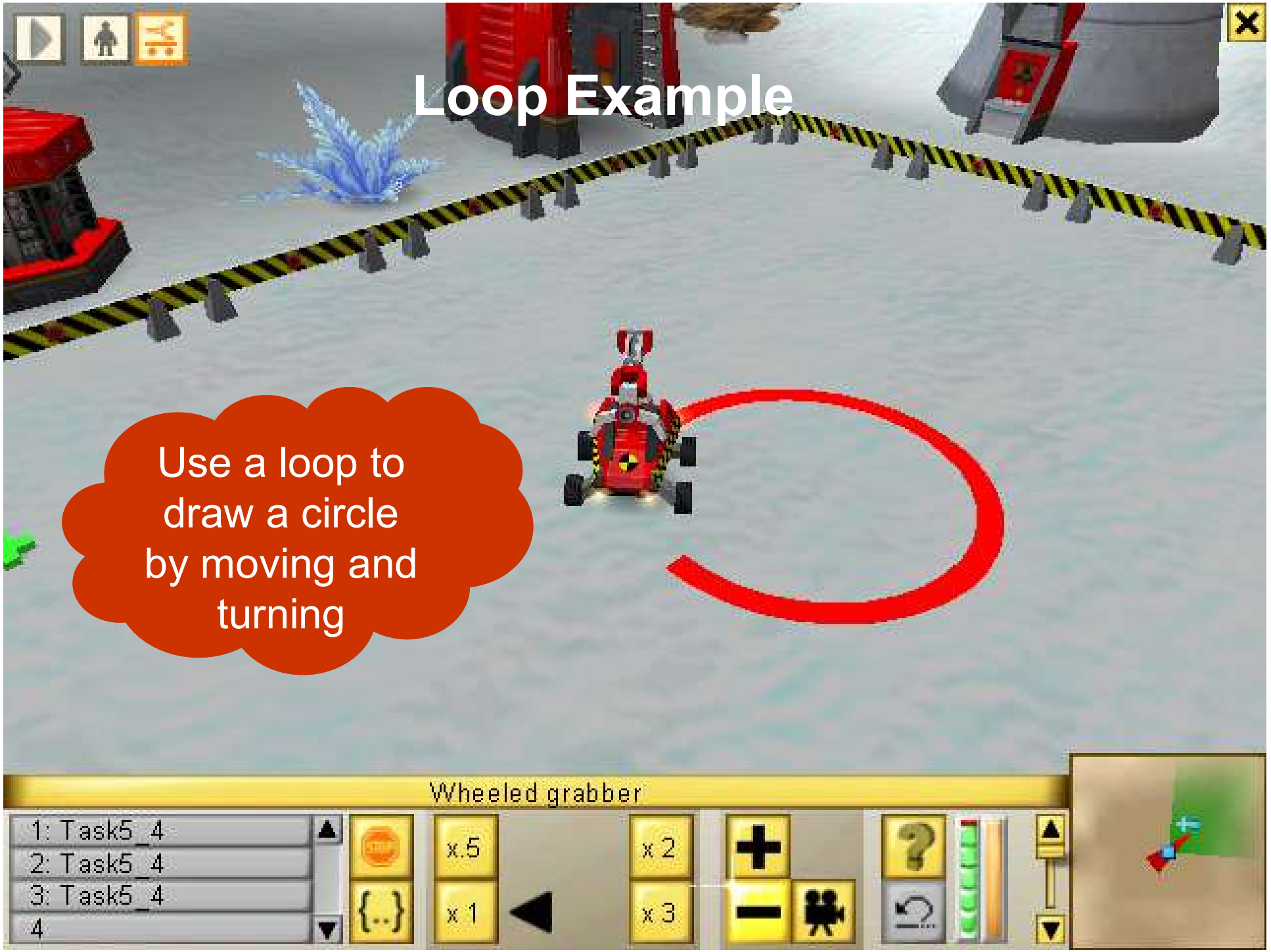
```
count = count + 1;
```

```
count --;
```



is equivalent to

```
count = count - 1;
```


Loop Example
Use a loop to draw a circle by moving and turning
Wheeled grabber
1: Task5_4
2: Task5_4
3: Task5_4
4
x.5
x 1
x 2
x 3

Example Program

Algorithm

1. Choose colour
2. Put pen down
3. Set count to zero
4. Loop while count < 36
   a. move 0.5 metres
   b. turn 10 degrees
   c. add 1 to count

   End loop

```
extern void object::DrawCircle()
{
    int count;          // declare int loop counter

    red();              // choose colour
    pendown();          // put pen down

    count = 0;          // initialise loop counter to zero

    while ( count < 36 )      // check whether condition is true
    {
        move (0.5);
        turn (10);
        count ++;       // add 1 to keep loop going

        message("Loop " + count + " Finished");
    }
}
```

What does this do?

for(…) loop

used to repeat
a block of code
a certain number of times

Syntax (grammar rules)

The **for(…)** loop does a similar job to the while loop

```
extern void object:: forLoop()
{
        //   the beginning of program goes here

          for ( initialise;  condition;  increment )
          {
              // block of code
              // to be repeated
              // goes here …
              //  block keeps repeating
          }

        //   the rest of the program continues when the loop finishes
}
```

Most work is done in 3 sections of the opening bracket

A for() loop Example

Display a message 10 times

```
extern void object:: Hello2()
{
        int count;      // set up a counter for the loop

        for (  count = 0 ;  count < 10 ;  count ++ )
        {
            message ("Hello Friends");
            wait (0.5);

        }

        message ("Loop Now Finished");
}
```

The result is a more compact loop

Circle Program 2

Algorithm 2

1. Choose colour
2. Put pen down
3. Loop 36 times
   a. move 0.5 metres
   b. turn 10 degrees
   End loop

```
extern void object::DrawCircle2()
{
    int count;          // declare int loop counter

    red();              // choose colour
    pendown();          // put pen down

    for ( count = 0 ; count < 36 ; count ++ )
    {
        move (0.5);
        turn (10);
        message("Loop " + count + " Finished");

    }

}
```

(count + 1)

This should be changed .. Why?

Alternative

Algorithm 2

1. Choose colour
2. Put pen down
3. Loop 36 times
   a. move 0.5 metres
   b. turn 10 degrees

   End loop

```
extern void object::DrawCircle2()
{
    red();
    pendown();

    for ( int count = 1 ; count <= 36 ; count ++ )
    {
        move (0.5);
        turn (10);
        message("Loop " + count + " Finished");
    }

}
```

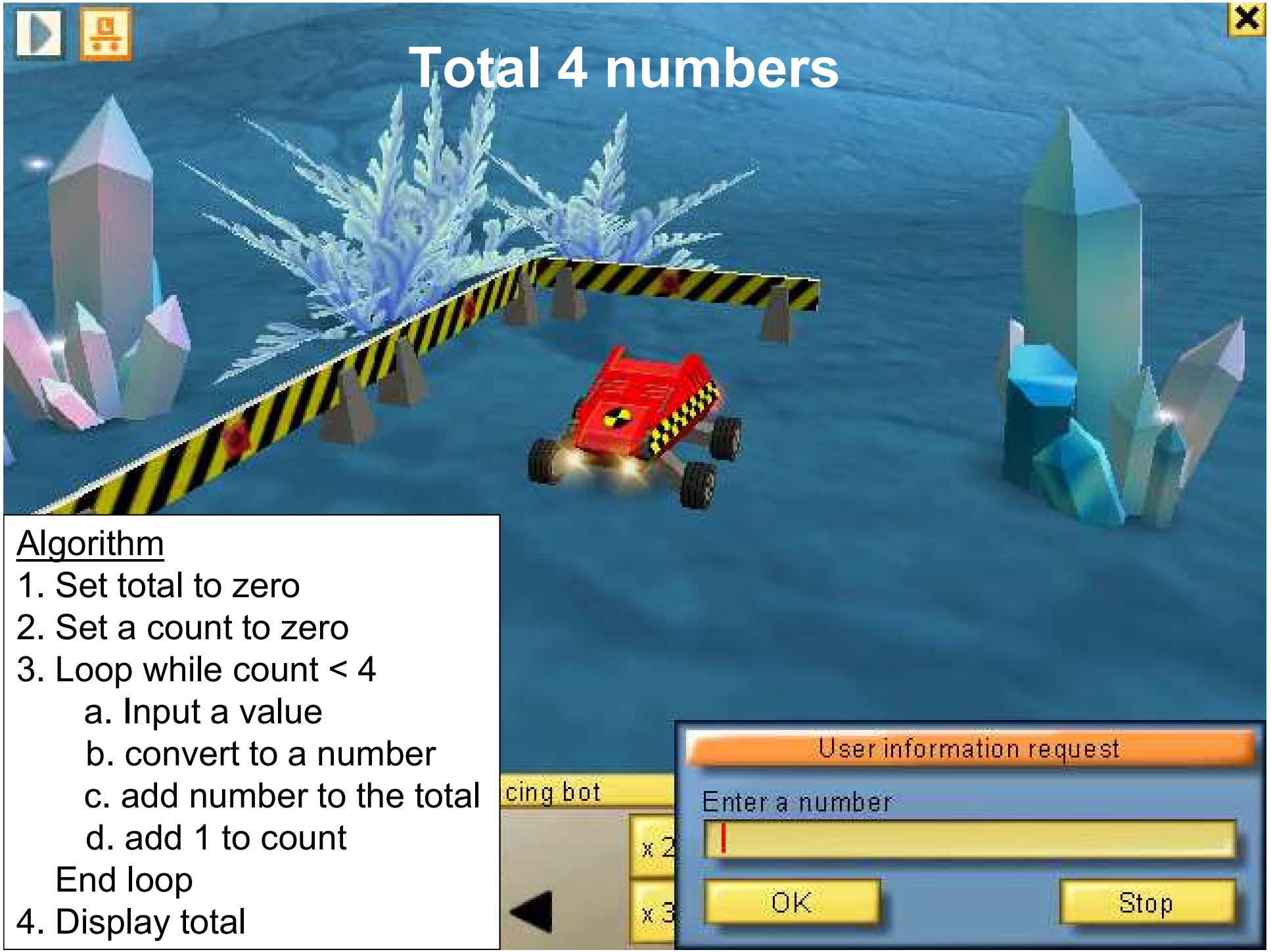
count declared here for use only inside the loop

count now starts at **1** and ends at **36**

What has been changed?

Using a loop to Input Numbers

Total 4 numbers



Algorithm

1. Set total to zero
2. Set a count to zero
3. Loop while count < 4
    a. Input a value
    b. convert to a number
    c. add number to the total
    d. add 1 to count
    End loop
4. Display total

Total Program

Algorithm
1. Set total to zero
2. Set a count to zero
3. Loop while count < 4
   a. Input a value
   b. convert to a number
   c. add number to the total
   d. add 1 to count
   End loop
4. Display total

```
extern void object::Totals()
{  // Author B N Ward : 18/12/2010
   // declare number & total, set total to 0
      float  number, total = 0;
      int  count = 0;      // initialise count to zero
      string input;        // declare a string for input

      while (  count < 4  )
      {
          input = dialog("Enter a number please");
          number = strval(input);     // convert input to number
          total = total + number;     // add number to total
          count ++;                   // add 1 to count
      }

      message ( "The total is " + total  );
}
```

Test Plan

A series of tests are planned and expected results calculated

Test No.	Input Numbers	Total	
		Expected	Actual
1	20, 40, 50, 60	170.00	
2	0, 0, 0, 0	0.00	
3	1.5, 2.5, 1.25, 2	7.25	
4	1, 2, 3, 4	10.00	
5	-1, -2.5, 0, 1.2	-2.30	

The program is tested and the actual results filled in here

Exchange Posts

used to supply information to robots


This is number 1 exchange post.
It holds the direction and distance (length) of the next post
Information exchange post
Number = 1.00
Direction = 0.00
Length = 50.00
SatCom
H

How to get information from an Exchange Post

The exchange post transmits information when the correct receive() instruction is used

using receive()

1. declare variables for angle and distance

```
float angle;
float dist;
```

2. receive direction and length information

```
angle = receive("Direction");
dist = receive("Length");
```

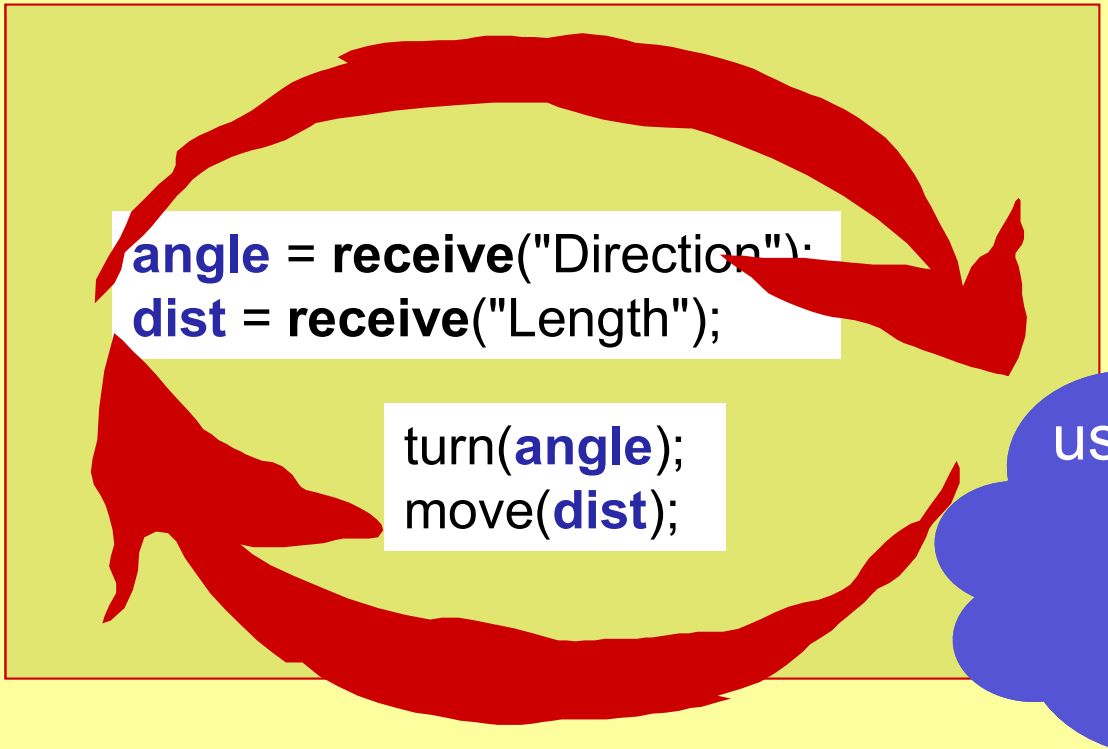
3. turn and move using values transmitted

```
turn(angle);
move(dist);
```

How can the robot complete the path?

It must repeat the process of:

1. receiving information
2. turning and moving to the next exchange post



using of course …
a while() loop
or a
for() loop


What I learned
1. iteration
2. while() loop
3. for() loop
4. conditions
5. relational operators
6. increment(++)
7. nested loops
8. Exchange Posts
9. receive()
THE END

Extra Reading

Nested Loops

Loops inside loops

How can we draw 4 Circles?

Algorithm

1. Choose colour
2. Put pen down
3. **Loop1** 4 times

    a. **Loop2** 36 times

        i. move 0.5 metres
        ii. turn 10 degrees

    End loop2

    b. Move 3 metres

End Loop1

inner Loop2 draws 1 circle

outer Loop1 repeats 4 times

move .. or circles will be on top of each other!

Draw 4 circles (using nested loops)

Algorithm

1. Choose colour
2. Put pen down
3. **Loop1** 4 times
   a. **Loop2** 36 times
      i. move 0.5 metres
      ii. turn 10 degrees
      End loop2
   b. Move 3 metres
   End Loop1

```
extern void object::Draw4Circles()
{
    red();              // choose colour
    pendown();          // put pen down

    for ( int outer = 1 ; outer <= 4 ; outer ++ )
    {
        for ( int inner = 1 ; inner <= 36 ; inner ++ )
        {
            move (0.5);
            turn (10);
        }

        move (3);
    }
}
```

inner loop draws 1 circle and is repeated 4 times