

CO452 Programming Concepts

Week 1 - Introduction to Ceebot,
Variables and Input/output



Aims and Objectives

Aim:

Introduce the Ceebot environment and apply the concept of variables

Learning outcomes:

- Learn about concepts of variables and inputting/outputting data
- Program solutions to small problems in Ceebot

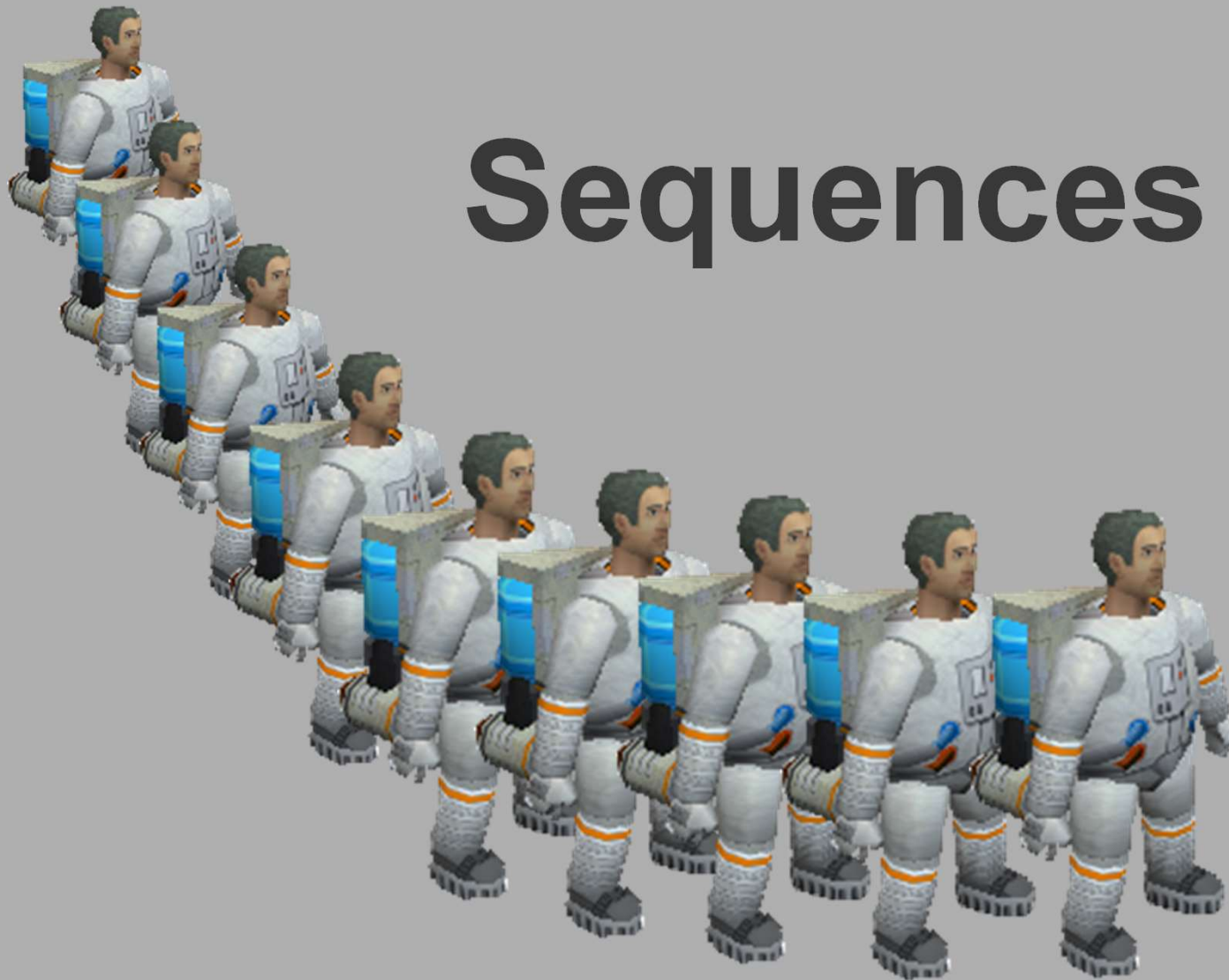
**What is a
computer program
???**



A Program is:

- **A set of instructions to the computer**
- **To make the computer do something useful**
- **Designed by programmers**
- **Written in a language like Ceebot, C++, Java**
- **Other high-level languages could be used .. e.g. Cobol, Fortran, Pascal, Basic, etc, etc.**

Sequences





Sequence

move(20);

grab();

turn(-90);

drop();

etc. etc.

The **sequence** is :

- a **block** of instructions .. one after the other
- with no deviation or repetition

The **order** of the instructions in a sequence is very important

if the order is changed, so is the logic of the program

The sequence is a basic construct of all programming languages



3 main constructs

Sequence, Selection, Iteration

These are constructs that are foundational to all programming languages

Algorithms



Why use algorithms?

- A plan for the program using english-like statements
- Algorithms can be used to design programs before coding starts
- This is especially important when writing larger and more complicated programs
- We use algorithms from the start, so you learn “good practice”



A basic algorithm example





Algorithm ... then Code

The destination is 20 meters away

the steps are
numbered in the
order of execution

Algorithm

1. Pick up item
2. Move 20 meters
3. Put down item



Program Code

```
extern void object::Task2_1()
{
    // Author: BWard. ID:156874
    // Course: BSc Comp
    // Date: 18/12/2010
    grab();
    move(20);
    drop();
}
```



Calling a function

function name *parentheses*

grab();



Functions that take a parameter

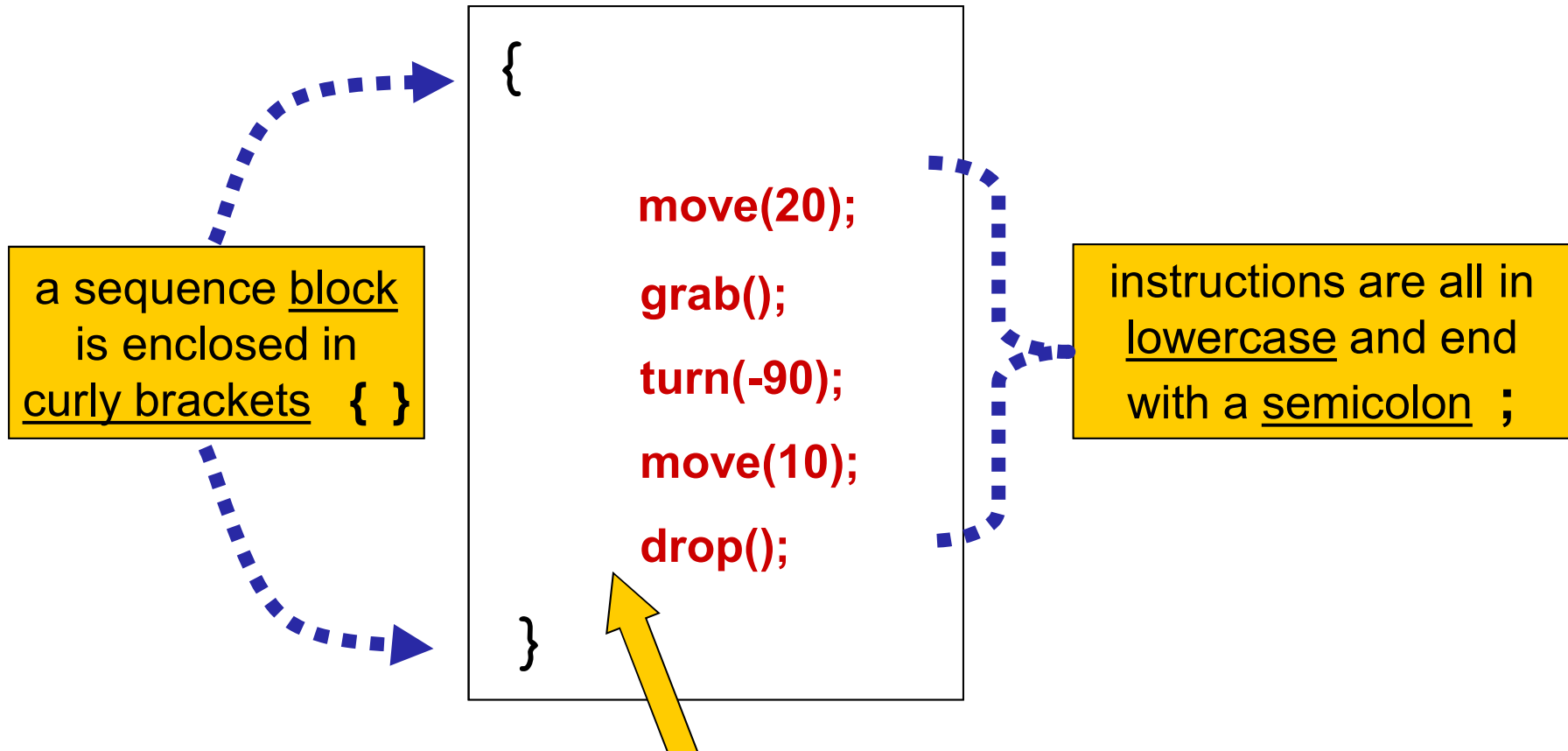
function name

Parameter

```
move(20);
```



How a Sequence of instructions is Programmed

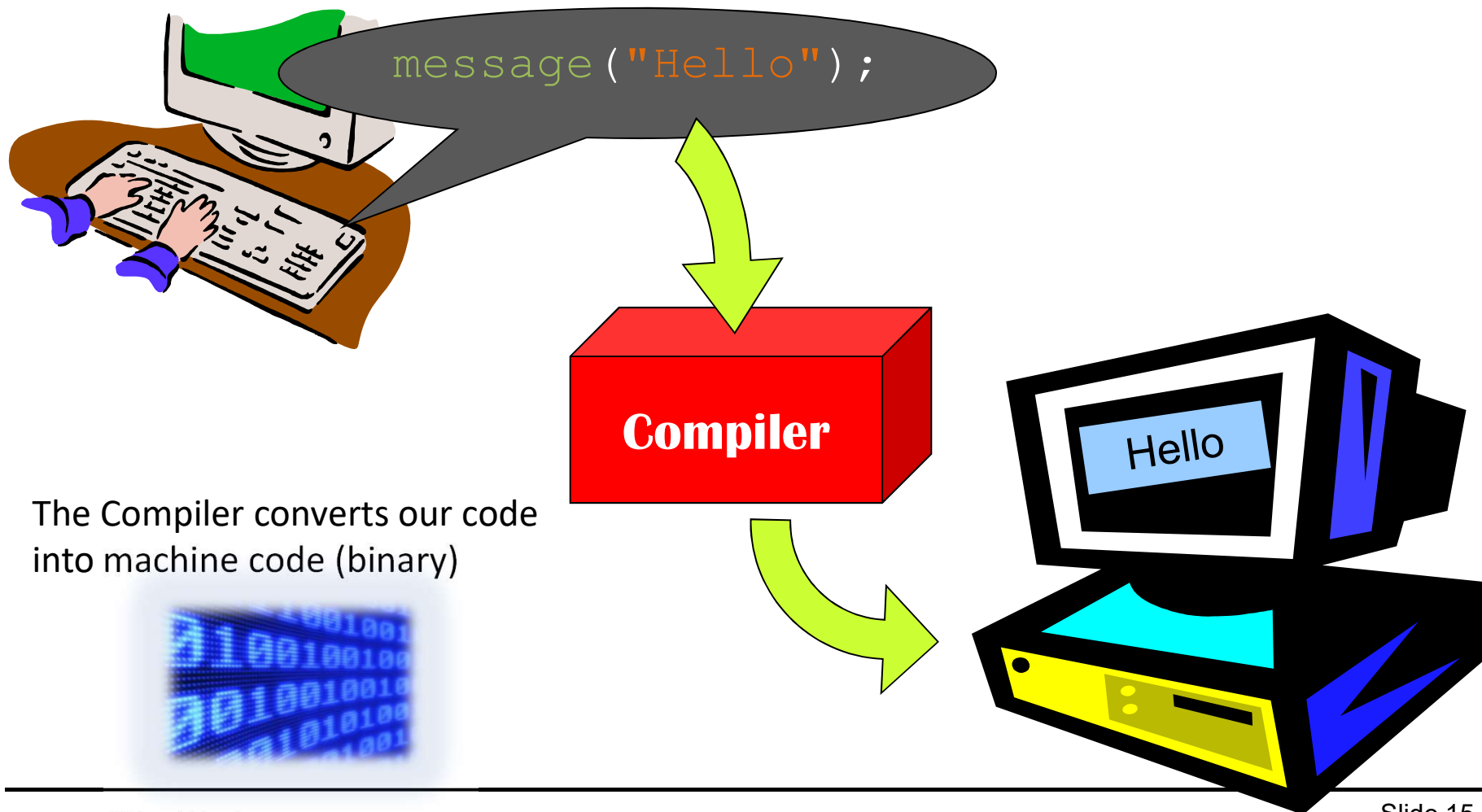


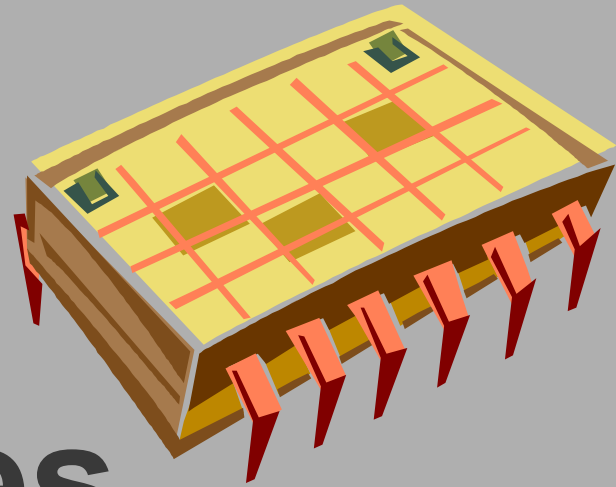
The instructions are **indented** (using tab key or spaces) and placed in the order in which they are to be executed (from top to bottom)



Computer Translation

How do computers understand our instructions?



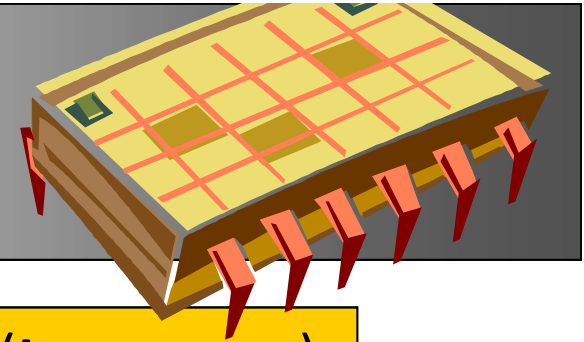


Variables

how to store information temporarily in a program



What is a Variable?



1. A storage area in the computer memory (**temporary**)
2. Can store information for use later in the program
3. A variable can be set up to store different types of data: numbers, words, etc.
3. The contents may change as the program runs (hence the name: variable)
4. Variables need to be given unique names
5. A variable name is also known as an identifier



Declaring a variable

type *identifier (or name)*

```
float width;
```



Assigning values to variables

```
float width;  
width = 20;
```

The value 20 would be stored in a memory address e.g. 002DCBF40
Compiler allocates a section of temporary memory to store this information
(rather than permanent memory)

The compiler substitutes the name we refer to as the variable for the memory address



A shorter way

```
float width = 20;
```

Note: this is also called initialising – giving a variable a value when it is declared



Assignment quiz

```
int a = 10;
```

```
int b = 20;
```

```
a = b;
```

What's the value of variable a?



Data Types for Ceebot Variables

There are 5 main data types for variables

int

Can store whole numbers e.g.

3 0 -261 46 -7

float

Can store numbers with decimal places e.g.

10.67 -0.05 13.0 176.4

string

Can store text (strings of characters) e.g. "High Wycombe" "Brian"

object

Can store details of an object e.g. Titanium, PowerCell

boolean

Can only be true or false

point

Can hold position coordinates

Each type needs a different amount of storage space



identifiers (variable names)

Rules for identifiers

1. The name must start with a letter
2. No spaces in the name
3. Can only have letters, digits, underscore
4. No reserved words (move, turn, etc.)
5. Length, length and LENGTH are all different variables
(i.e. Ceebot is case-sensitive)
6. Good Practice:
always choose meaningful names

Name OK or not?

My_Name	✓
my-name	✗
1stname	✗
D2	✓
Number4	✓
%cost	✗
first name	✗



What is Ceebot?



An Animated Virtual World of Robots





3 elements of programming

variables
sequence **iteration**
 arrays
selection objects

Programming concepts



Syntax



Environment

How to Use Ceebot



Ceebot Start Screen

CeeBot4

CeeBot4 COLLEGE CAMPUS 1.3.005 E, © Epsitec SA 2001-2008
The site license for this COLLEGE version has been granted to:
Buckinghamshire New University, United Kingdom..
This program must be used only on the campus of this institution.

Select your class:

class1

Cancel

Select your name:

Student
Teacher

OK



The Main Ceebot Menu

Select 'Standard'

Programming exercises

Exercise series:
 College Standard Additional User levels

Chapters:

1: Introduction	
2: Sequences & Algorithms	x
3: Using Variables	x
4: Input and Output	x
5: Loops 1	x
6: Selection	x
7: == Extra Exercises ==	x

Exercises in the chapter:

1: Landing	x
2: Move a Robot	<input checked="" type="checkbox"/>
3: Moving and Turning	x
4: Move an Object	x
5: Move Backwards	x
6: Power up a Robot	x
7: Killer Wasps	x

Summary:
Learn how to move a robot forwards

Quit Options Play

Start the exercise



Click on the
Wheeled grabber

Wheeled grabber

Teacher



x.5
x 1



x 2
x 3





Click here to go to the code editor

Edit the selected program



Program editor

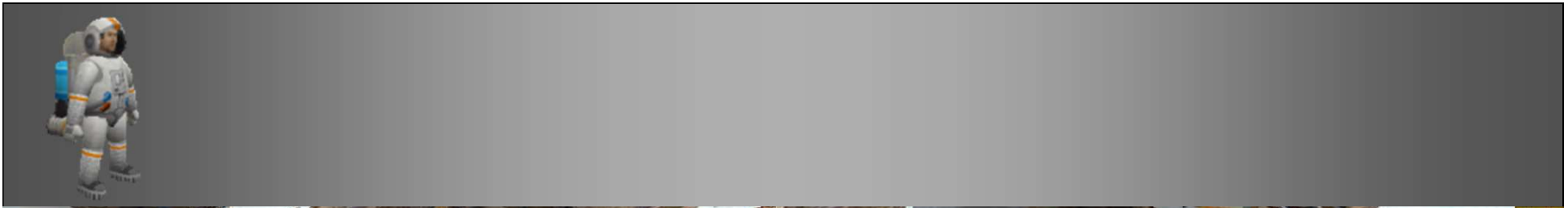
extern void object: Task1_2 ()
{
move (20) ;
...
}

This statement moves the robot forward 20 meters

Click here to exit the editor

OK Cancel

↓ → ⤴ 1



Wheel Grabber

1: Task1_2	▶	x.5	x 2	+	?	▲			
2	◀								
3									
4	▼	{..}	x 1	◀	x 3	-	👤	15	▼

Execute the selected program

Click here to run the program

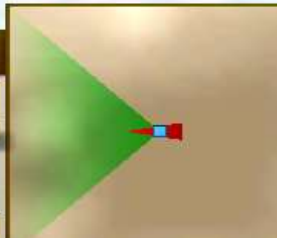


<<< Well done, mission accomplished >>>



Wheeled grabber

1: Task1_2	▲	→	x.5	x 2	+	?	▬	▲
2	▬							
3	▼	{..}	x 1	x 3	-	🎥	↺	▼
4								





Question

Could you rewrite this solution
to include a variable?



Program editor

File Edit Undo Cut Copy Paste Run Help ?

```
extern void object: Task1_2 ()  
{  
    int len = 20;  
    move (len) ;  
}
```

OK Cancel Run Stop 1



Activity

Attempt exercise 1 in the study pack (Task 4.1)





```
int len = 20;  
move(len);  
turn(90);  
move(len);  
turn(-90);  
move(len);  
turn(-90);  
move(len);
```

Output

*using message(...) instruction
to display information
on the screen*



Using message(...) with various parameters

Assume

```
string myname = "Brian Ward";  
int age = 35;
```

```
message("Hello World");
```

Hello World

```
message("Hello " + myname);
```

Hello Brian Ward

```
message(myname + " is " + age + " years old.");
```

Brian Ward is 35 years old.

message(...) uses + to join the parts to form one output string



Example Output



So I am Robbie and you are Brian
... that's so ****COOL****

2 message(...) outputs





Maths: Calculate total cost of some items

```
extern void object::Task5_3()
{
    // declare variables needed by program
    float price, total;
    int quantity;

    // put values into variables
    price = 10.50;
    quantity = 20;

    // work out total value of goods
    total = price * quantity;

    // output the result
    message("The total price is " + total + " pounds");
}
```



In other languages

C# `Console.WriteLine("Hello World");`

C++ `cout << "Hello World" << endl;`

Java `System.out.println("Hello World");`

Input

*using dialog(...) instruction
to input information from
the keyboard*



Using dialog(...) to input words

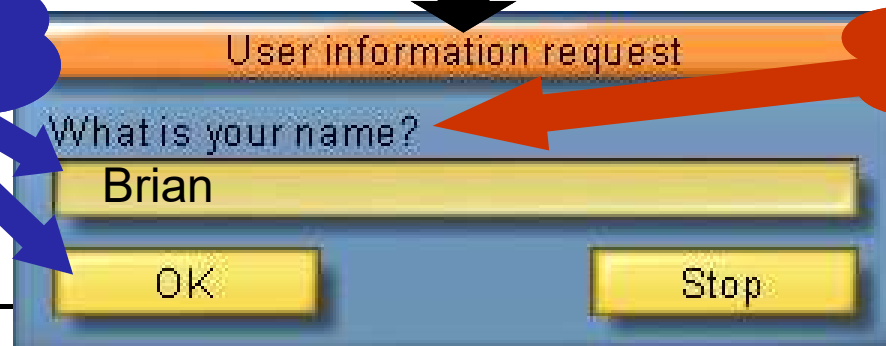
1. First declare a string variable to hold the input

```
string username;
```

2. Use this to gather the information using dialog(...)

```
username = dialog ( "What is your name?" );
```

Enter and
click OK



user prompt
parameter



Example Program

```
extern void object::Task6_1()
{
    string username;           // declare a string variable called username

    move(18);                   // move robot forward
    wait(1);                    // wait for 1 second

    // input user's name
    username = dialog("Human, please tell me your name?");
    wait(1);

    // output a message using the name
    message("Thank you" + username + "I am very pleased to meet you.");

    // better version
    message("Thank you " + username + ", I am very pleased to meet you.");
}
```

What's wrong?
(hint: spaces)



Thank you Brian, I am very pleased to meet you.



Tracked grabber

1: roboChat	▲		x.5	x 2
2				
3			x 1	◀ x 3
4	▼			

User information request

Human, please tell me your name?

Brian

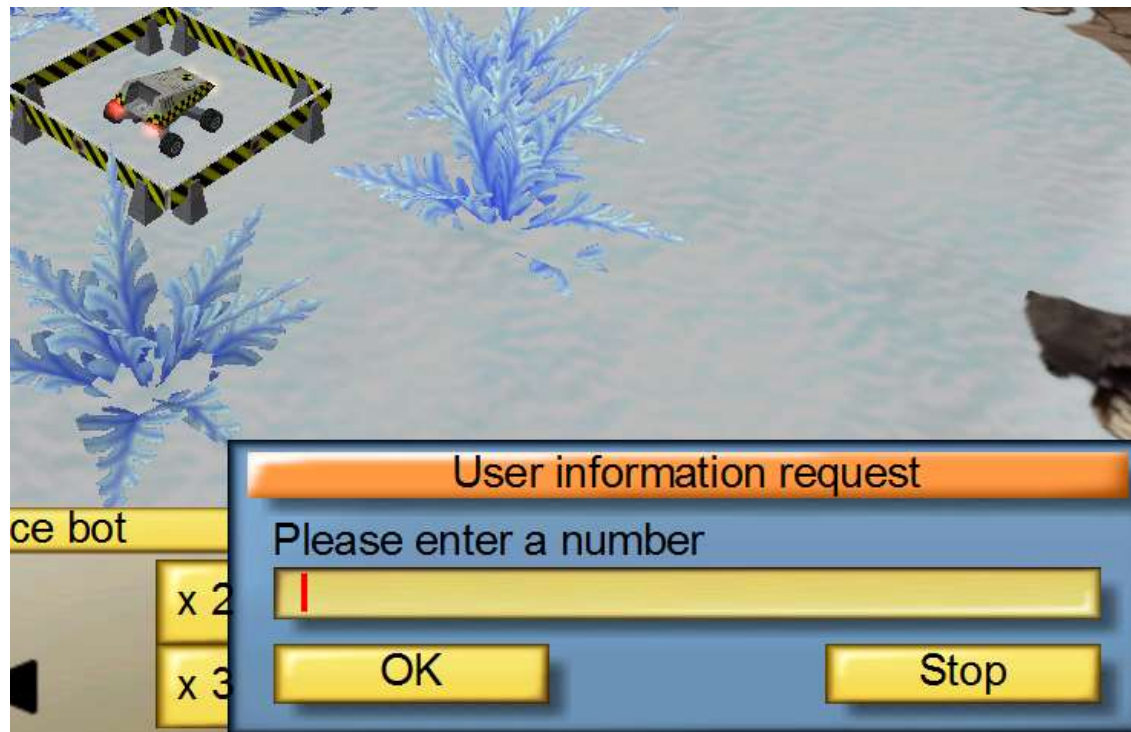
OK

Stop



Activity

Attempt exercise 2 in the study pack (Task 5.3)



Program comments



Good Practice

Use comments in your programs

```
// this is a one line comment  
// the compiler ignores this line
```

```
/* this is a multiple-line  
comment and the compiler will  
ignore everything here  
*/
```

Use comments to:

- identify author and program
- explain trickier parts

Example Uses

```
extern void object::Task0_7()  
{  
    /* Author: B Ward  
    Date: 14/08/05  
    Task: install power cell */  
  
    grab();  
    turn(-90); // turn clockwise  
    wait(0.5); // pause for 0.5 sec  
    drop();  
}
```



Quiz!

What is the correct way of declaring a integer variable with the identifier 'x'?

a. `x int;`

b. `x;`

c. `int`

d. `int x;`



Quiz!

What are the three main constructs found in all programming languages?

Sequence, Selection, Iteration



Lecture: [finished]

What we looked at today:

- Algorithms
- Sequence
- Variables
- Input/output

Extra Reading



Push [F1] to get instructions

sat Com

◀ ▶ 🏠 A A A A 🖨️ 📄

Task 1.1 : Destroy an Alien

You are an astronaut.
Your spaceship has just landed on an alien planet.
First of all you need to look around.
Use the arrowkeys on the keyboard to move forward and examine the scene below you.
You should see a robot shooter and a threatening alien ant.
The robot is programmable from a distance. You need to program the robot so that it destroys the alien (see below for howto do this).
Notice that the shooter robot is already lined up correctly, facing the ant, so all you have to do is fire .. at the right moment!

Skipping the introductory movie

If you want to have another go at any exercise, just hit the reset button .
But if you don't want to watch the movie again, hit the [esc] key when the movie starts.

How to enter and execute a program

Click with the mouse on the robot in order to select it:



🔌

Return to the exercise



Assignments

Information can be stored in a variable using:

the assignment statement
and assignment operator (=)

e.g:

age = 25 ;

wage = 15.50 ;

choice = "A" ;

name = "Brian Ward" ;

title = "Menu List" ;

Computer Memory

Variable	Contents
age	25
choice	A
wage	15.50
name	Brian Ward
title	Menu List



Some Useful Instructions

fire(...);

move(...);

turn(...);

grab();

drop();

wait(...);

message(...);

pendown();

red();

**Put them in the right order
and use the correct
parameters to create
your program.**

=====

**each instruction ends with
a semicolon ;**



Which of these instructions have parameters?

✓
move(20);

✓
fire(1);

grab();

pendown();

✓
wait(0.5);

drop();

red();

✓
turn(90);

Note: most instructions have brackets, but not all use them



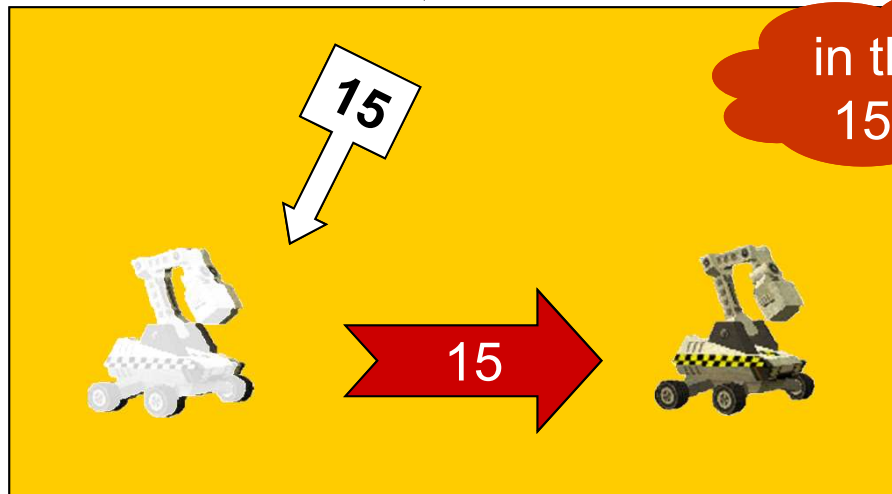
Parameters

Brackets () act like a doorway into the instruction

15
↓

the parameter is
a value passed
in

move(**15**) ;



in this case,
15 is used

The parameter is used to complete the instruction: **move(15);**