Introduction to Computers and Programming

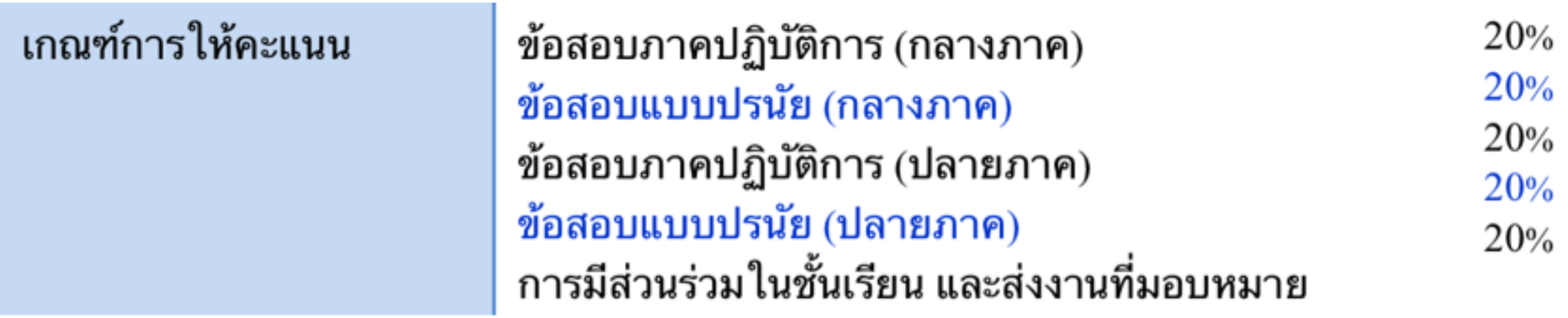
01204111 Computer and Programming

Department of Computer Engineering
Kasetsart University

Cliparts are taken from http://openclipart.org

Outline

- Introduction to computer programming.
  - A bit of computer anatomy and a quick history of computing.
- Examples of programming concepts from code.org.
- A quick look at the C# programming language.



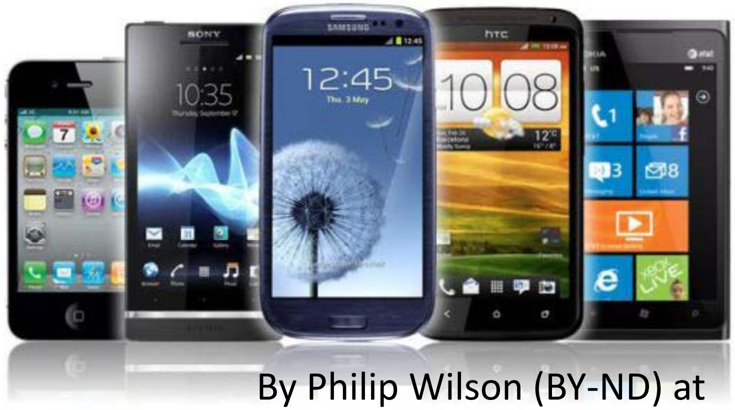
เกณฑ์การให้คะแนน	ข้อสอบภาคปฏิบัติการ (กลางภาค)	20%
	ข้อสอบแบบปรนัย (กลางภาค)	20%
	ข้อสอบภาคปฏิบัติการ (ปลายภาค)	20%
	ข้อสอบแบบปรนัย (ปลายภาค)	20%
	การมีส่วนร่วมในชั้นเรียน และส่งงานที่มอบหมาย	20%

The age of computing

- Computers are everywhere.

By Marc van der Chijs at https://www.flickr.com/photos/chijs/21798665468

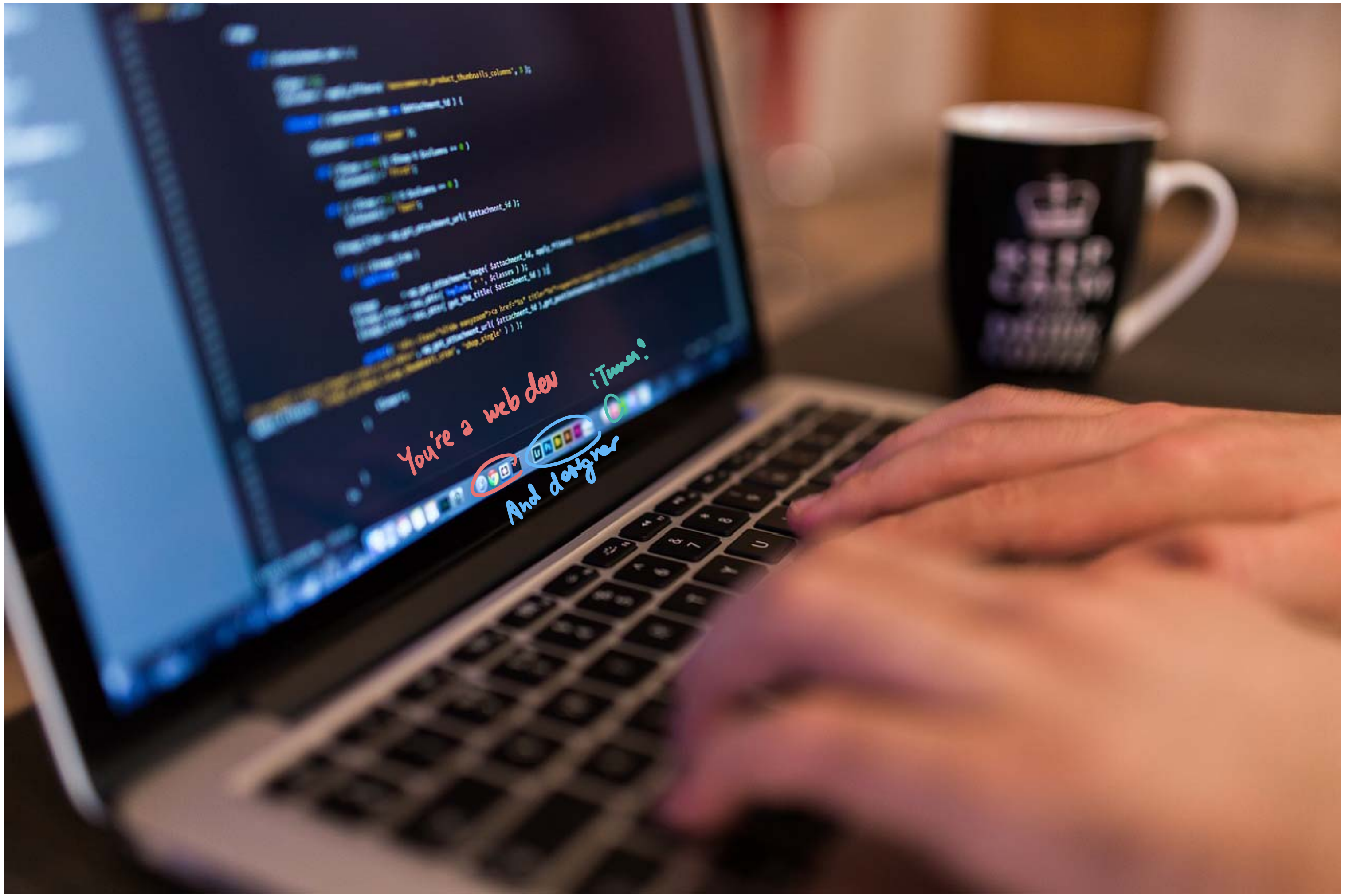
From: https://pixabay.com/en/network-iot-internet-of-things-782707/



By Philip Wilson (BY-ND) at https://www.flickr.com/photos/internetsense/9900738813/

Computer programming

- Programming – an act of developing computer programs.
- What is a computer program?



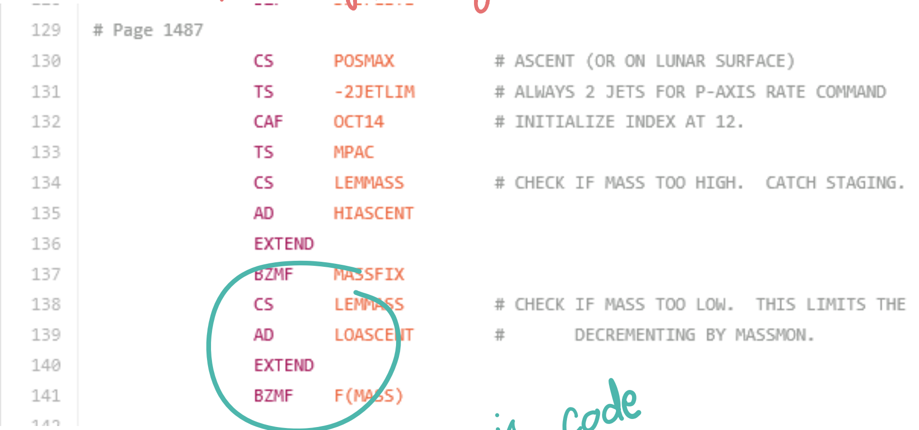
Public domain Image from https://pixabay.com/en/work-typing-computer-notebook-731198/

A computer program

dot matrix.

- Margaret Hamilton with the computer program that took Apollo 11 to the moon.
- You can read the code at: https://github.com/chrislgarry/Apollo-11

A very, very new slide.



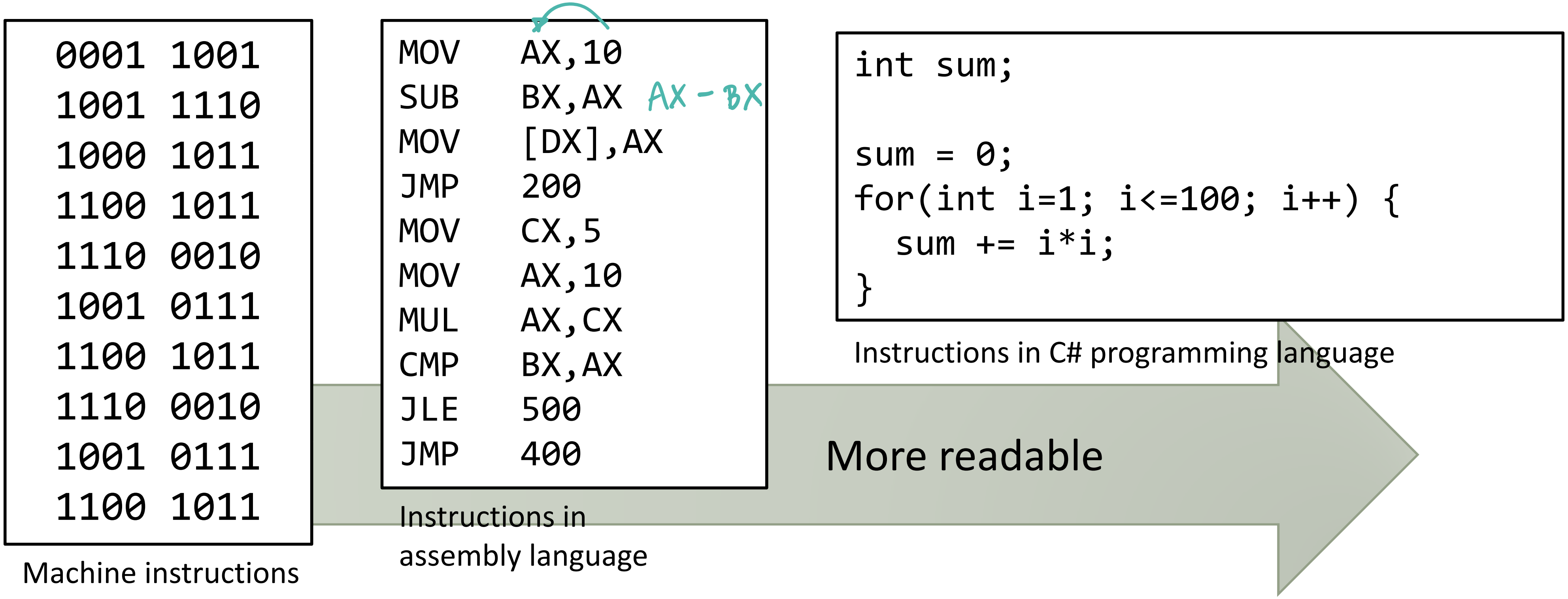
```
# Page 1487
                CS      POSMAX          # ASCENT (OR ON LUNAR SURFACE)
                TS      -2JETLIM        # ALWAYS 2 JETS FOR P-AXIS RATE COMMAND
                CAF     OCT14           # INITIALIZE INDEX AT 12.
                TS      MPAC
                CS      LEMMASS         # CHECK IF MASS TOO HIGH.  CATCH STAGING.
                AD      HIASCENT
                EXTEND
                BZMF    MASSFIX
                CS      LEMMASS         # CHECK IF MASS TOO LOW.  THIS LIMITS THE
                AD      LOASCENT        #       DECREMENTING BY MASSMON.
                EXTEND
                BZMF    F(MASS)
```

mnemonic code

Public domain image from https://en.wikipedia.org/wiki/Margaret_Hamilton_(scientist)#/media/File:Margaret_Hamilton.gif

A computer program

- A **computer program** is a sequence of **instructions** to be executed by computers.
- Examples of computer programs in various forms:



```
0001 1001
1001 1110
1000 1011
1100 1011
1110 0010
1001 0111
1100 1011
1110 0010
1001 0111
1100 1011
```

Machine instructions

```
MOV    AX,10
SUB    BX,AX  AX - BX
MOV    [DX],AX
JMP    200
MOV    CX,5
MOV    AX,10
MUL    AX,CX
CMP    BX,AX
JLE    500
JMP    400
```

Instructions in assembly language

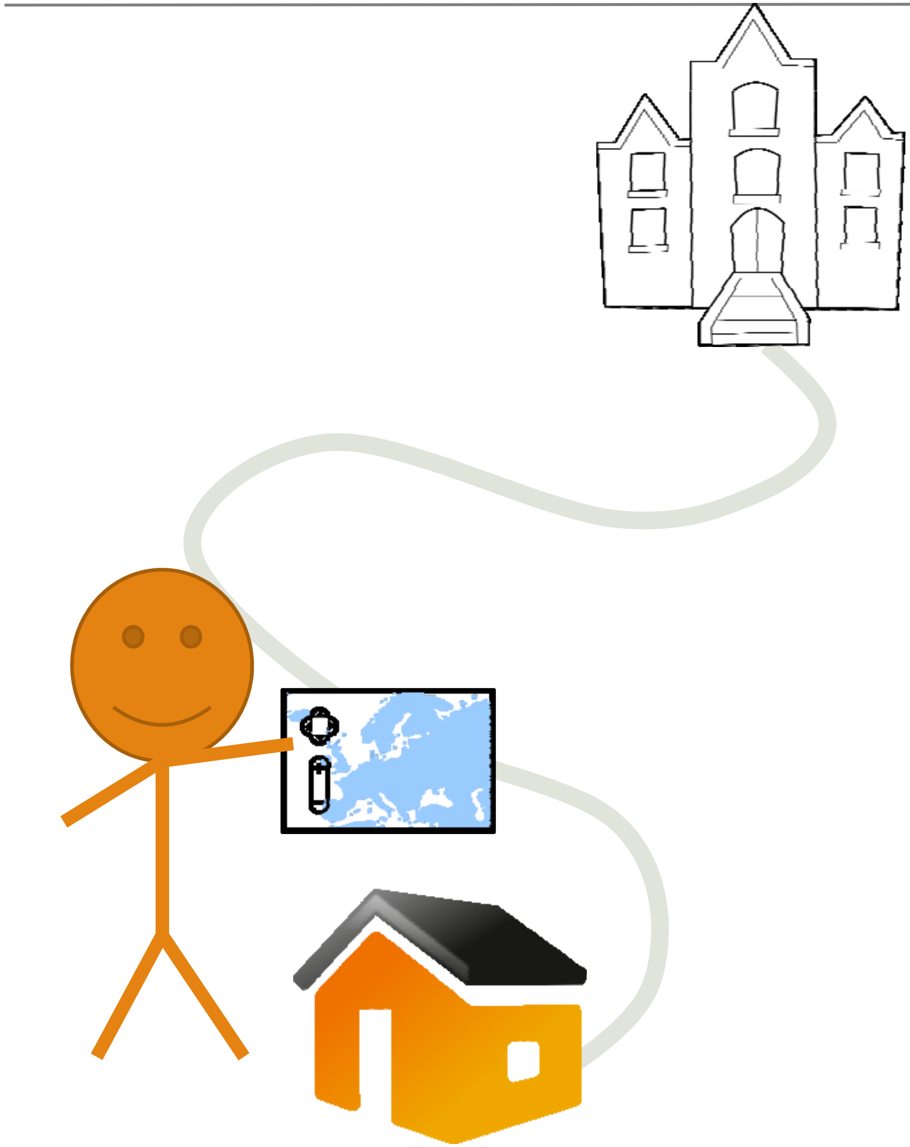
```
int sum;

sum = 0;
for(int i=1; i<=100; i++) {
  sum += i*i;
}
```

Instructions in C# programming language

More readable

From home to school



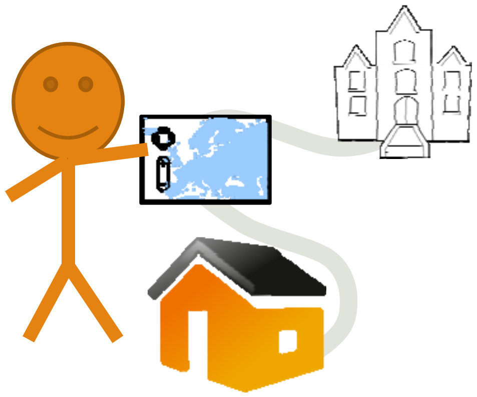
- To understand how computer work, let's try to make an analogy with how people solve some problem.

A specific problem

- **Problem:** It's the first day of school. You want to go to KU from your home. What do you have to do?
  - Assume that your home is close to KU, so you decide to walk to KU.

Is KU counted as school?

Walking from home to school



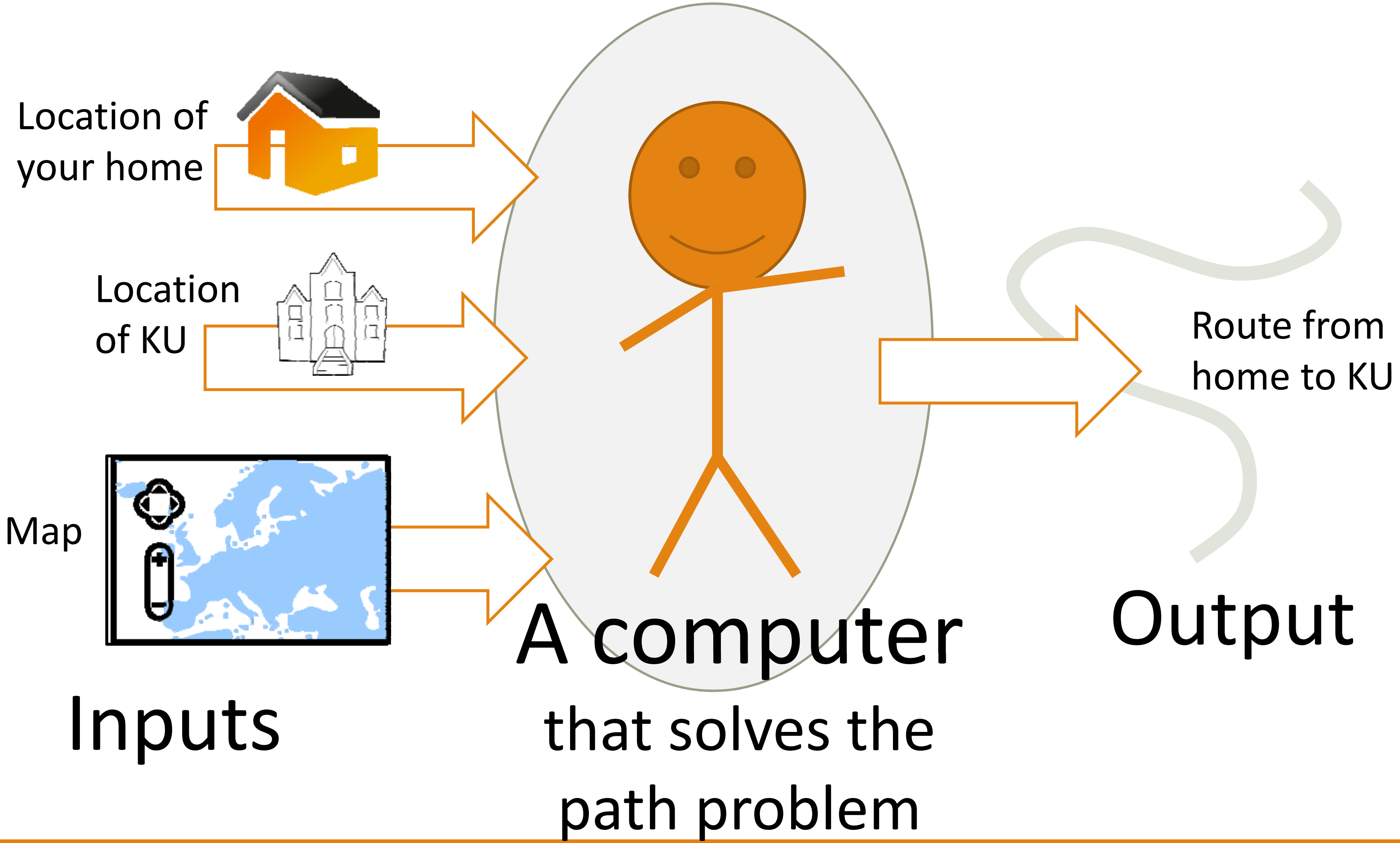
- If you know the way to KU, you can just walk. But if you don't you may want to look at **the map** and use it to plan your route to KU.

- Note that if you can **plan your walking route with a map**, you can solve this kind of problems, ☆☆☆ not just for going from your home to KU, but from any place to any other place.

Non-specific problems.

A computer, inputs, and outputs

- In a way, you are a computer.



Location of your home

Location of KU

Map

Inputs

A computer

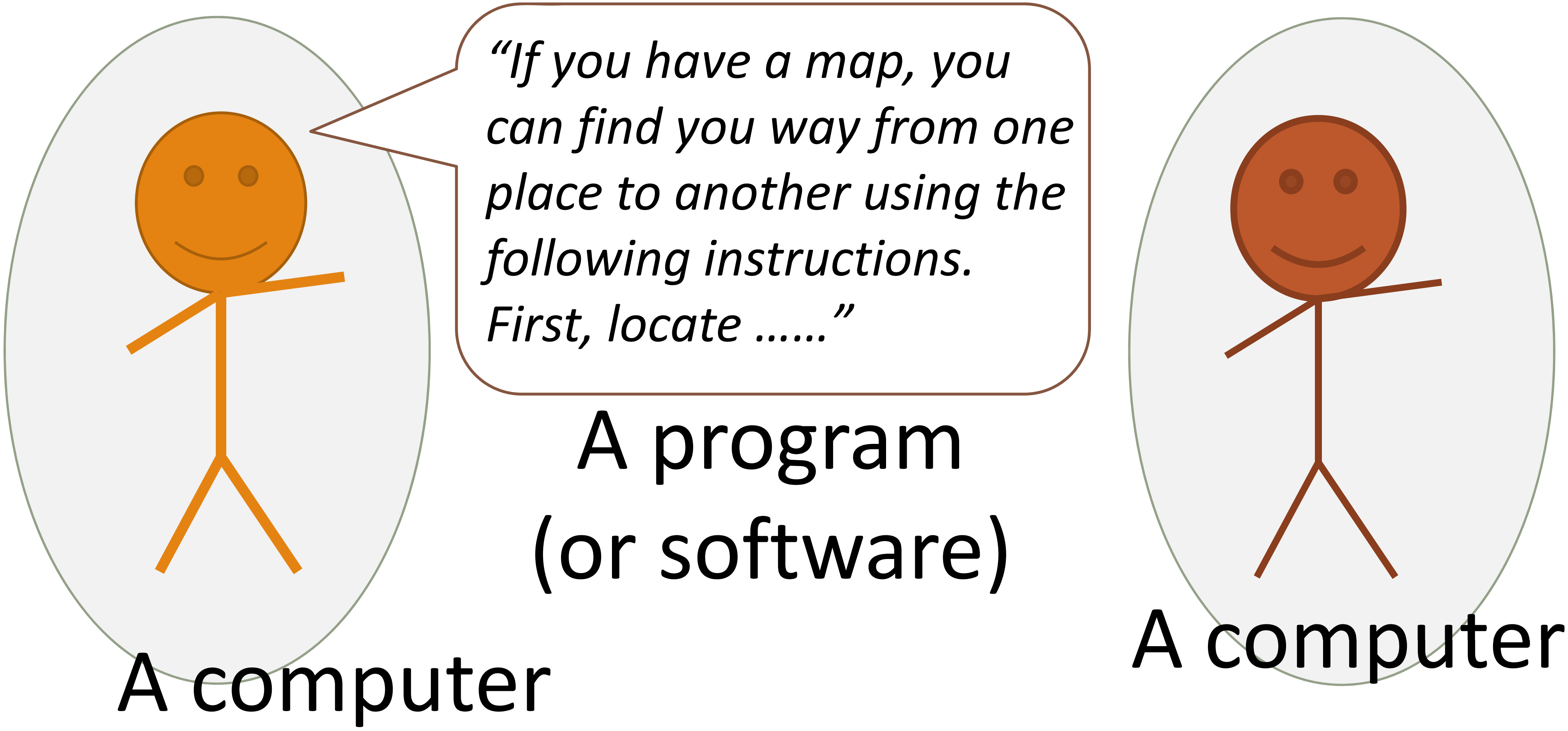
that solves the path problem

Route from home to KU

Output

A program

- Can you teach other people to solve the same problem?



“If you have a map, you can find you way from one place to another using the following instructions. First, locate ……”

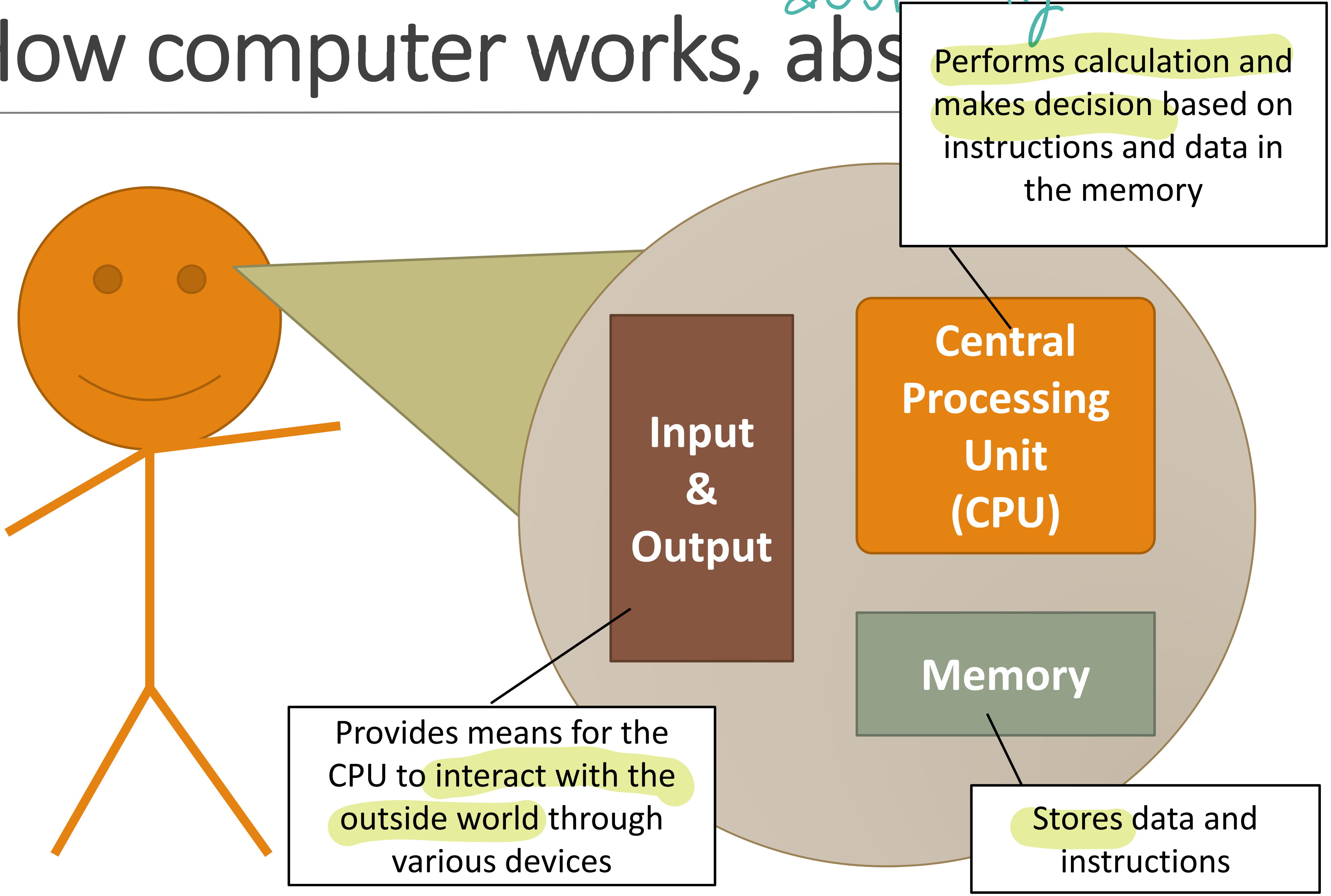
A program (or software)

A computer

A computer

abstractly

How computer works, abs

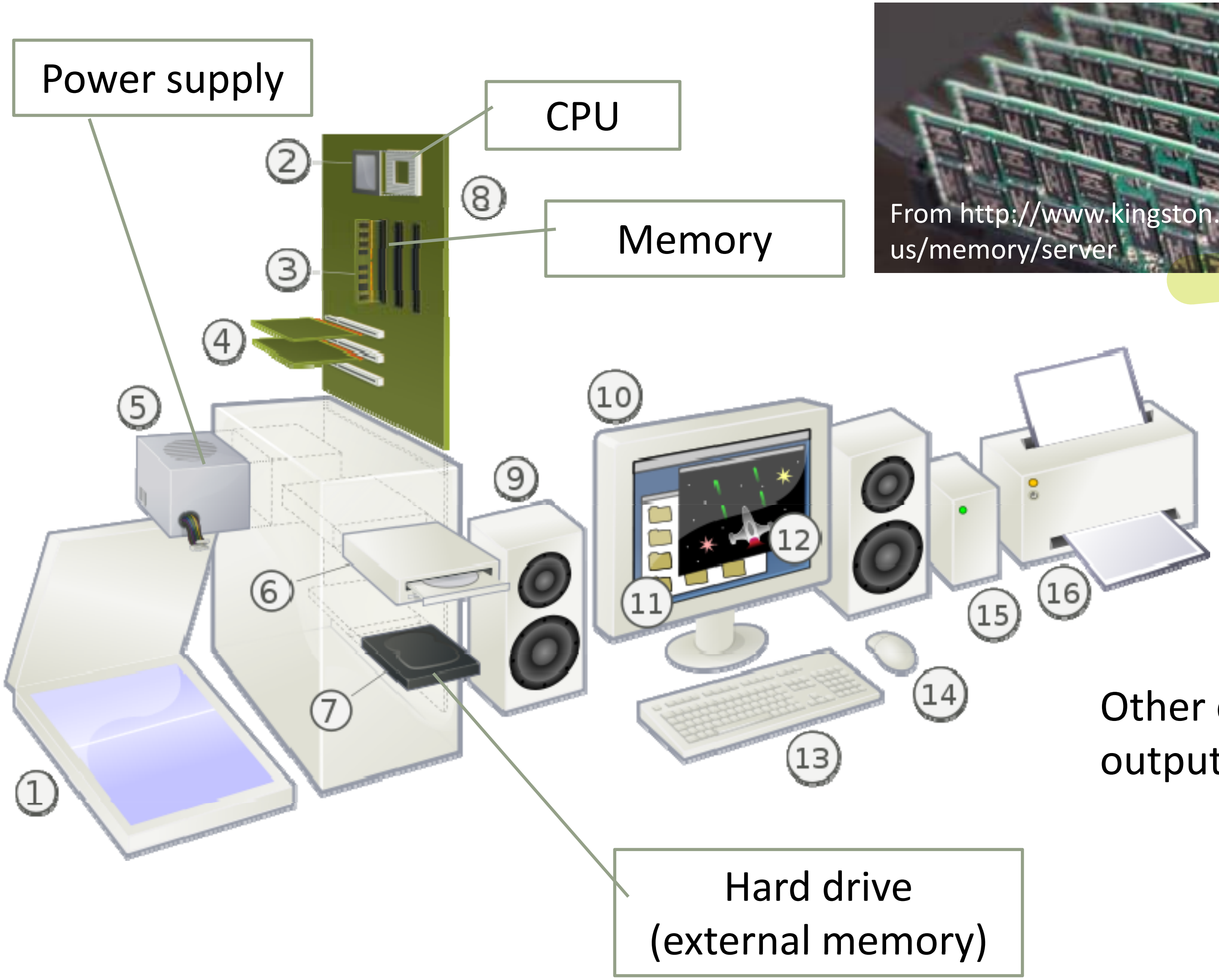


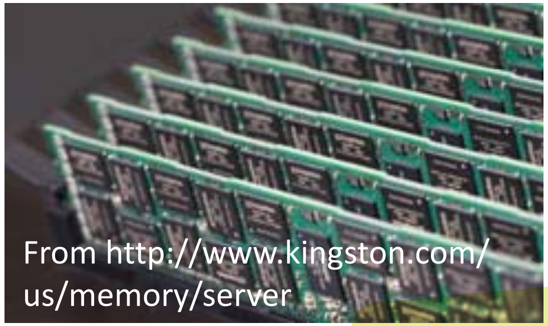
Performs calculation and makes decision based on instructions and data in the memory

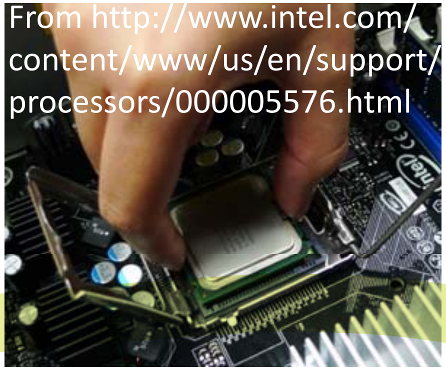
Provides means for the CPU to interact with the outside world through various devices

Stores data and instructions

The real computer components




From http://www.kingston.com/us/memory/server


From http://www.intel.com/content/www/us/en/support/processors/000005576.html

Other devices are input or output devices

Main Image by user HereToHelp
from https://en.wikipedia.org/wiki/File:Personal_computer,_exploded_6.svg

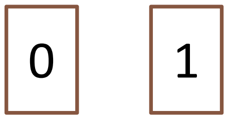
Yes, your smartphone is a computer too

- If you disassemble your smartphone, you will find CPU(s), memory units, and other I/O devices as well.

CC-BY-SA Image by Tyler Love, from iPhone Repair
URL: http://flickr.com/photos/28004346@N08/3399597800

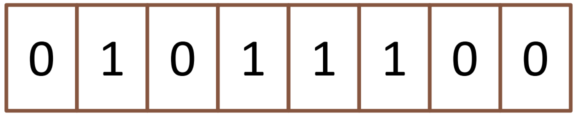
Inside the memory

- The smallest unit of information that can be processed by digital computers is a single binary digit (a **bit**), which can be either 0 or 1.
- We usually group them in groups of 8 bits, each called a **Byte**.
- A lot of bytes can be stored in a memory unit.
  - 1 kB = 1,000 bytes
  - 1 MB = 1,000,000 bytes
  - 1 GB = 1,000,000,000 bytes



0 1

Two bits

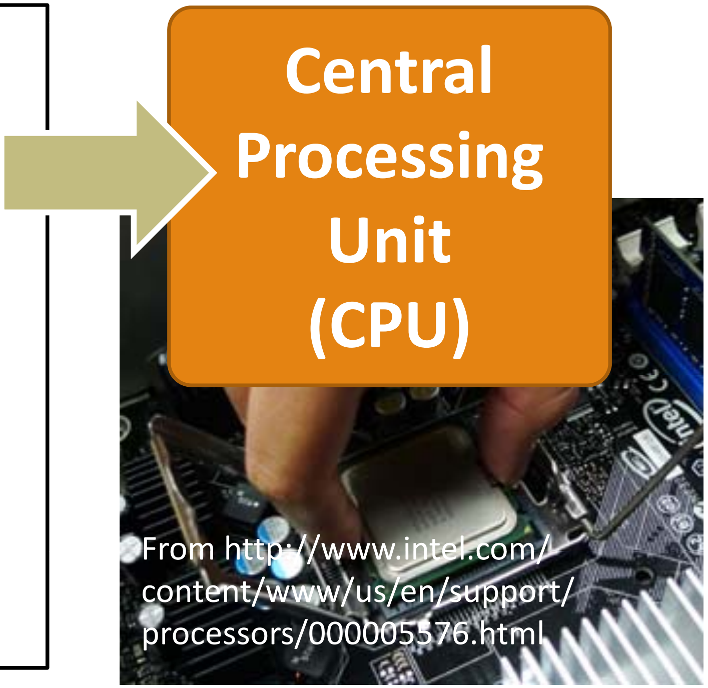


0 1 0 1 1 1 0 0

One byte

The instructions

```
0001 1001
1001 1110
1000 1011
1100 1011
1110 0010
1001 0111
1100 1011
1110 0010
1001 0111
1100 1011
```



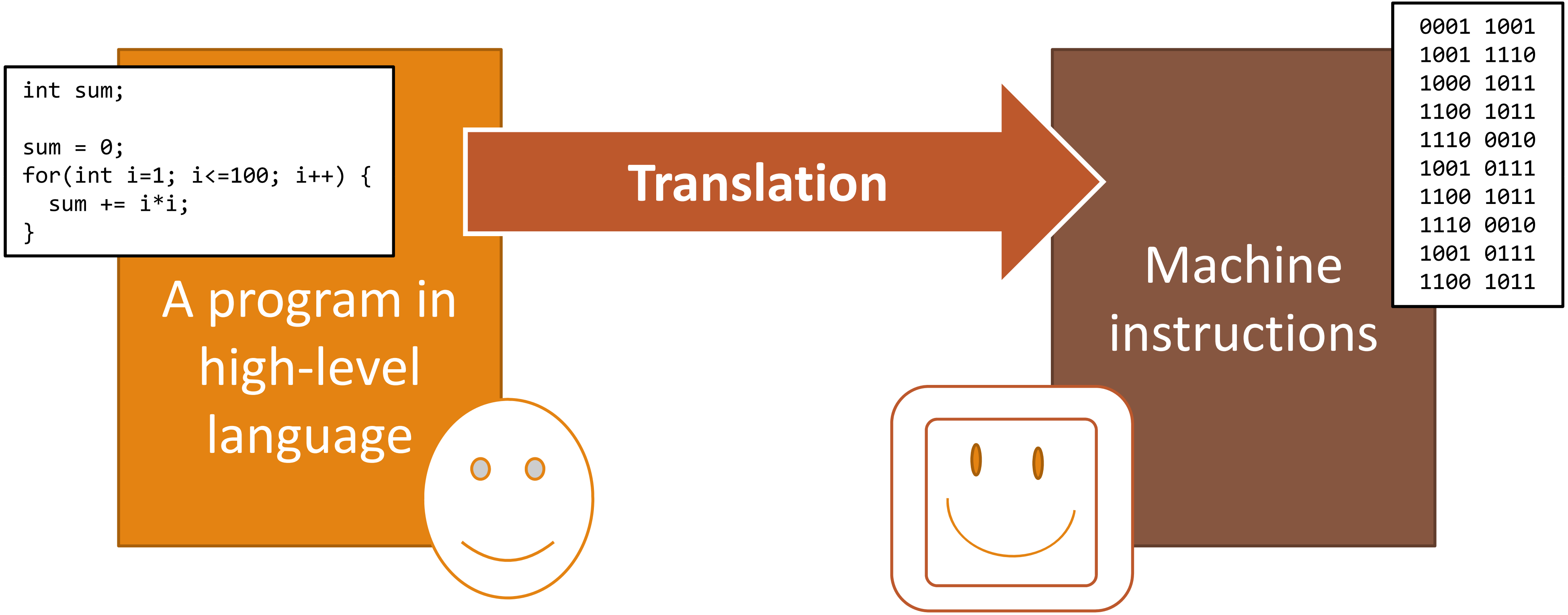
Central Processing Unit (CPU)

From http://www.intel.com/content/www/us/en/support/processors/000005576.html

- The memory, not only keeps the data to be processed with the CPU, but it also keeps the instructions.
- These instructions are in the format that the **CPU** can easily understand, referred to as **“machine instructions.”**
- When writing a program, we rarely write in machine instructions.

From programs to instructions

- Instead of working directly with machine instructions, people usually develop software with higher-level programming languages.
- But the program must be translated into a form that the computer can understand. This process is called **program translation**.



```
int sum;

sum = 0;
for(int i=1; i<=100; i++) {
  sum += i*i;
}
```

A program in high-level language

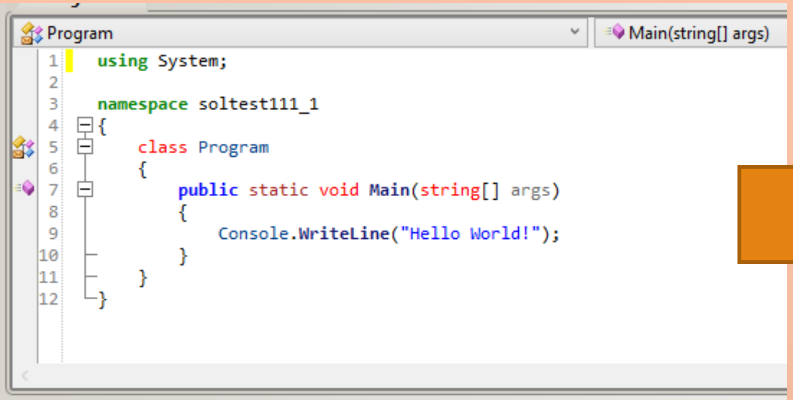
Translation

Machine instructions

```
0001 1001
1001 1110
1000 1011
1100 1011
1110 0010
1001 0111
1100 1011
1110 0010
1001 0111
1100 1011
```

Compilers

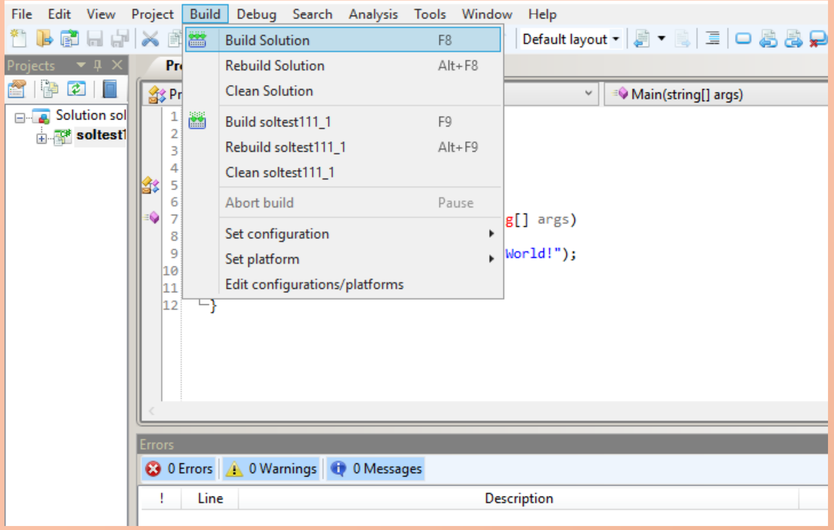
- There are many ways a program can be translated into a machine-readable form. For the programming language used in this course, C#, a special software called a **compiler** performs that task.



```
using System;

namespace soltest111_1
{
    class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

We write a program in C#.



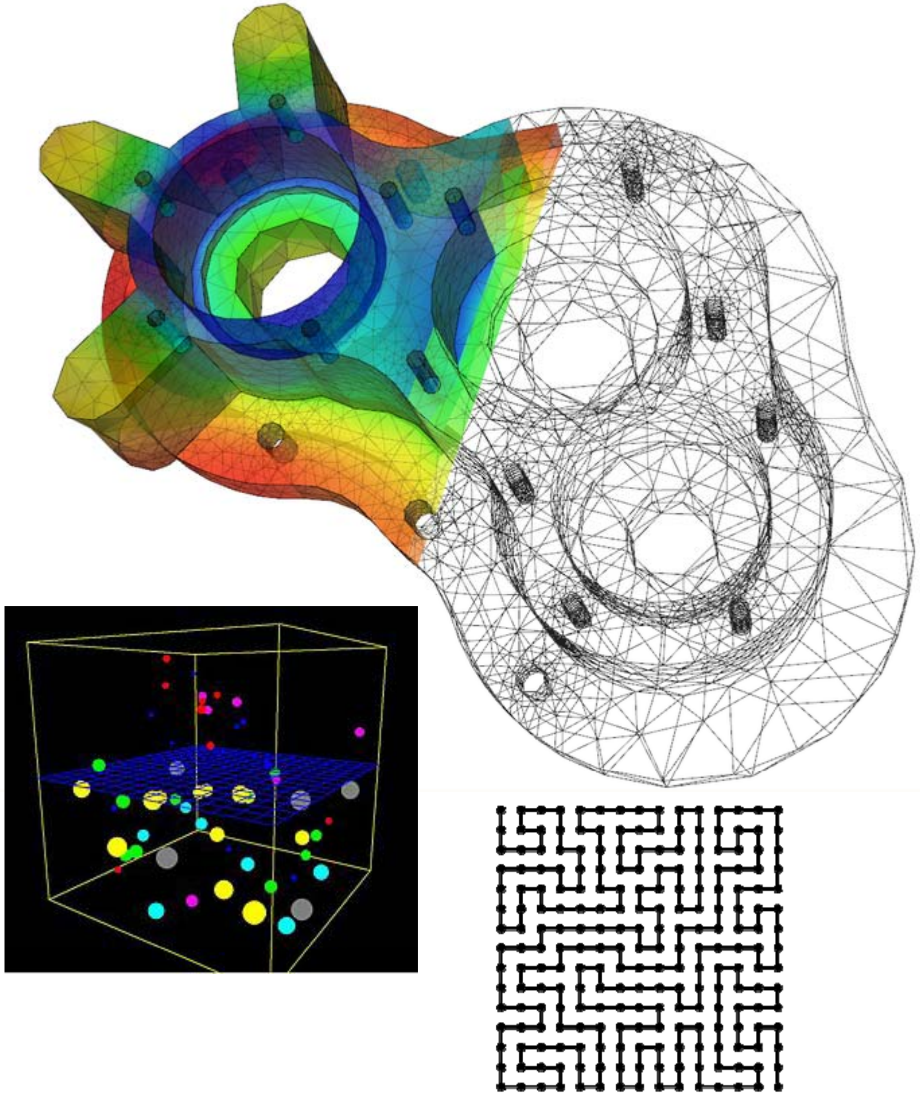
The program has to be built by a compiler, before you can run it.

Why do you want to learn how to program?

- Computer programming is not the easiest thing to learn, but it will definitely be useful to you.

From: https://pixabay.com/en/computer-female-girl-isolated-15812/ (CC0 license)

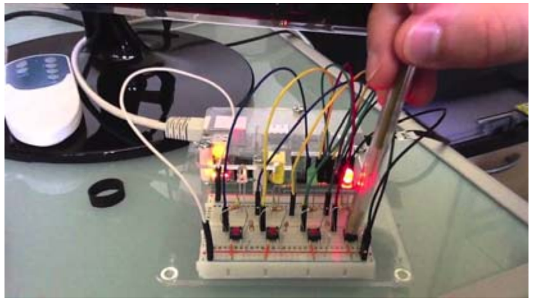
For your career

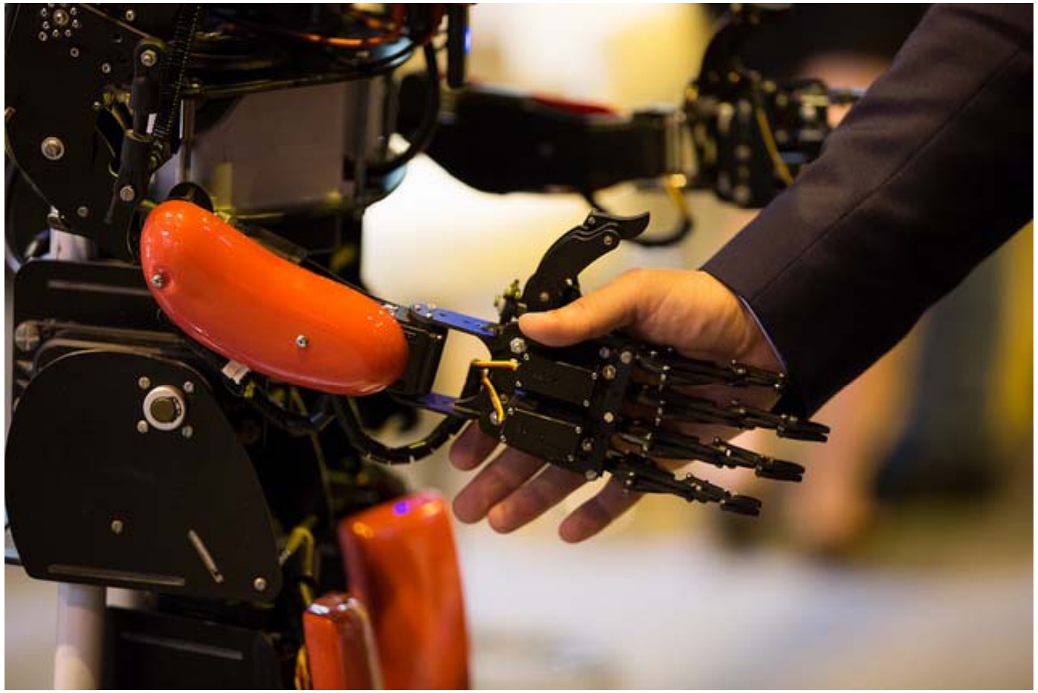


- As an engineer, you will have to perform lots of important computation tasks.
- Knowing how to program gives you advantages:
  - you can write the programs to do these tasks yourself, or
  - if you let someone develop programs for you, you might have a better judgement on the quality of the work.

(1) By User A1 from https://commons.wikimedia.org/wiki/File:Elmer-pump-heatequation.png (CC-BY-SA)
(2) By User Lazarus666 from https://commons.wikimedia.org/wiki/File:Osmosis_computer_simulation.jpg
(3) By Rocchini from https://commons.wikimedia.org/wiki/File:Self_avoiding_walk.svg (CC-BY-SA)

It is central to innovations





- Many exciting innovations have components that perform intelligent tasks.
- They usually rely on powerful software running on the devices.
- With recent cheap prototyping hardware boards, innovators can try new ideas faster by writing codes on existing hardware platforms.

(1) From: https://www.youtube.com/watch?v=H_xmR35Ws0w (2) By Multicherry From: https://commons.wikimedia.org/wiki/File:Raspberry_Pi_2_Model_B_v1.1_front_angle_new.jpg (CC-BY-SA) (3) By L'Ecole polytechnique at https://www.flickr.com/photos/117994717@N06/28066156056

Tech startups

- If you want to build a tech startup that changes people's life, it is very important that you know how to code so that you can implement your ideas quickly and create values.

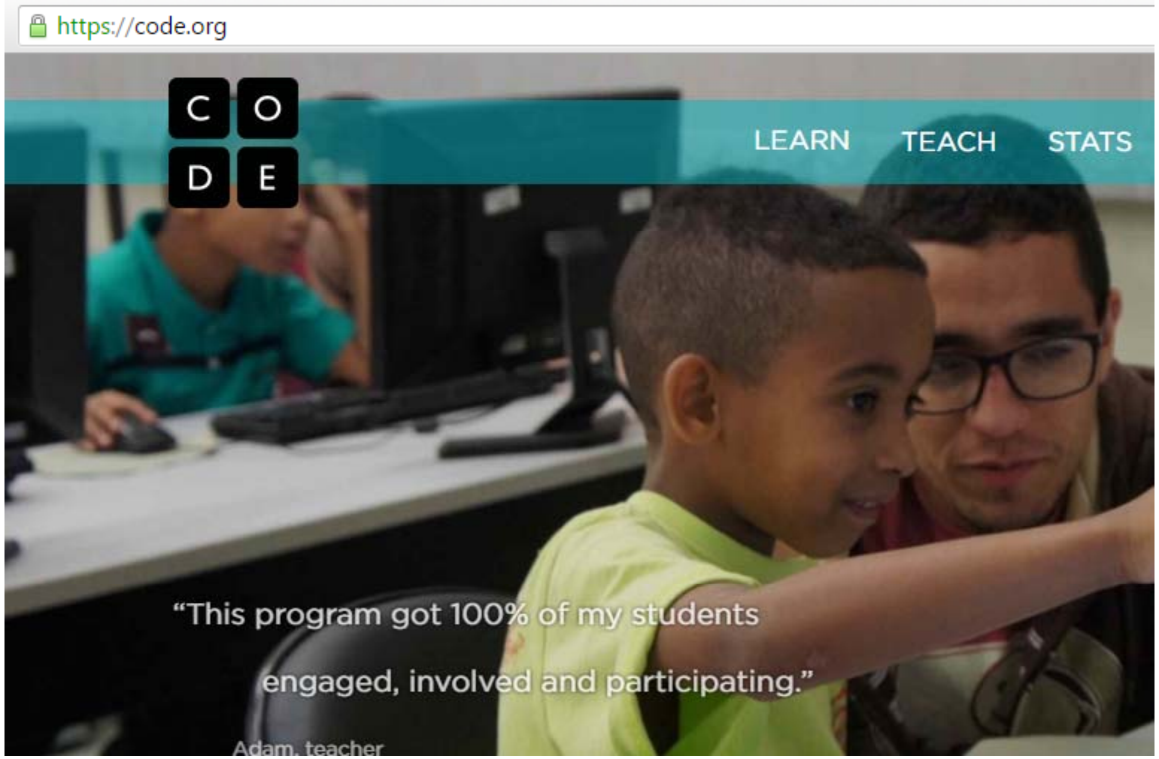
By Jisc and Matt Lincoln from https://www.jisc.ac.uk/rd/get-involved/supporting-technology-startup-projects (CC-BY-NC-ND)

Finally, it’s fun

From: https://pixabay.com/en/children-win-success-video-game-593313/ (CC0 license)

Quick start: programming with blocks

- We will start learning how to program by looking at many simple programming exercises from the site: code.org.
- It’s a great website for learning how to program.

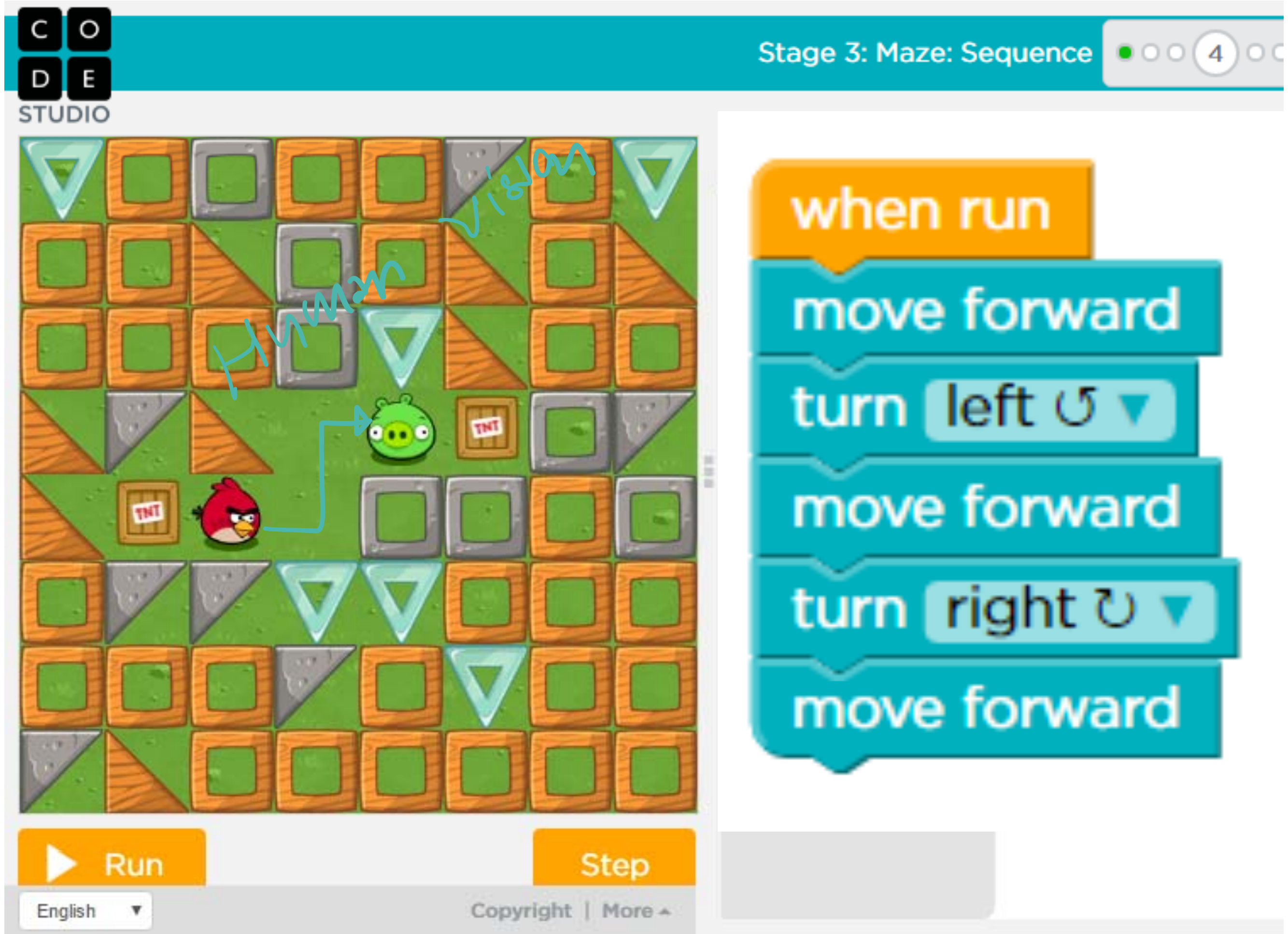


https://code.org

“This program got 100% of my students engaged, involved and participating.”

Adam, teacher

Reaching goals

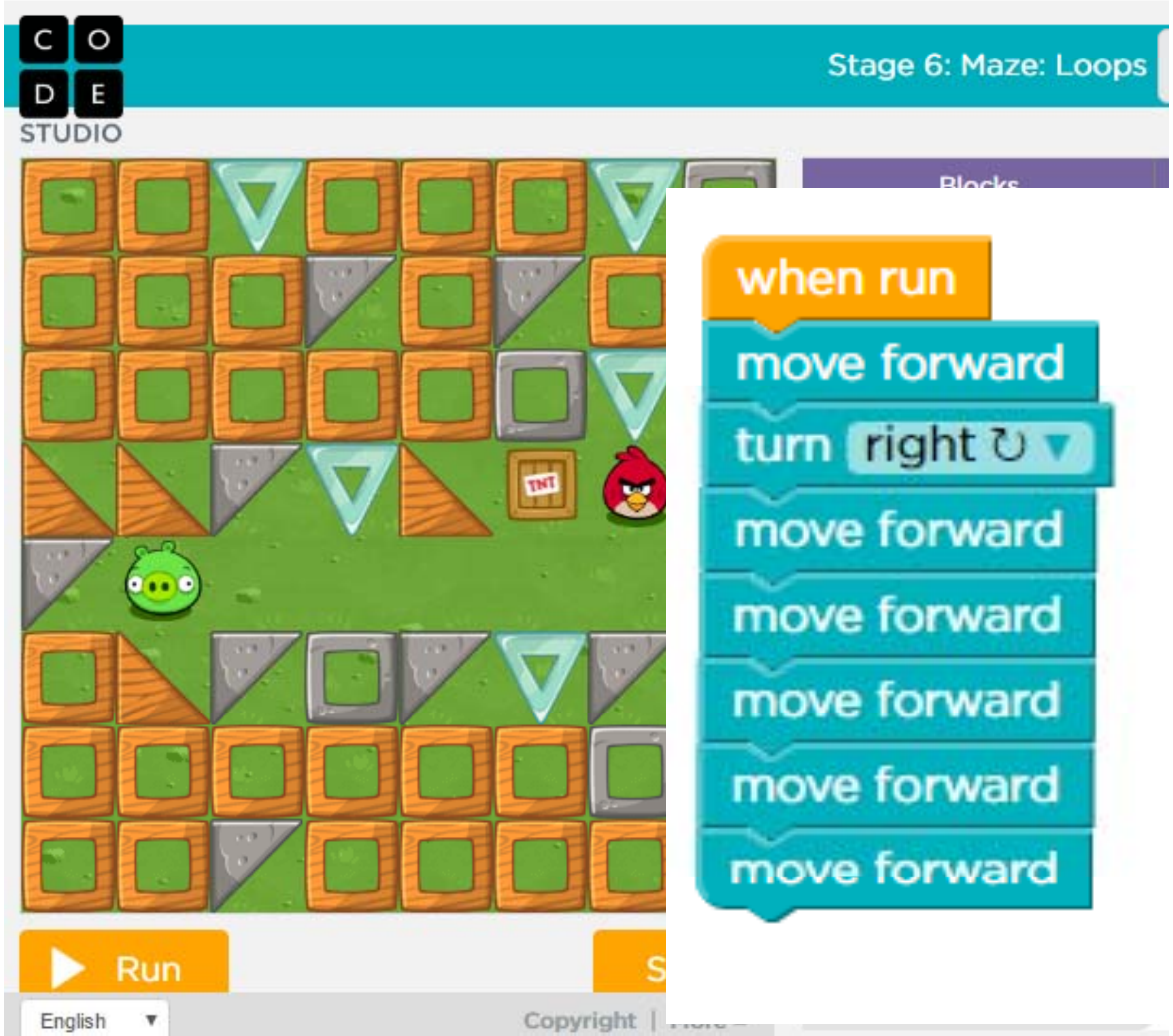


```
when run
move forward
turn left ↺
move forward
turn right ↻
move forward
```

- You (the red bird) want to hit the pig.
- Possible instructions that you can use are “move forward”, “turn left” and “turn right.”
- What is the sequence of instructions that you need?

From: https://studio.code.org/s/course2/stage/3/puzzle/4 (CC-BY-NC-SA License)

A longer sequence (1)

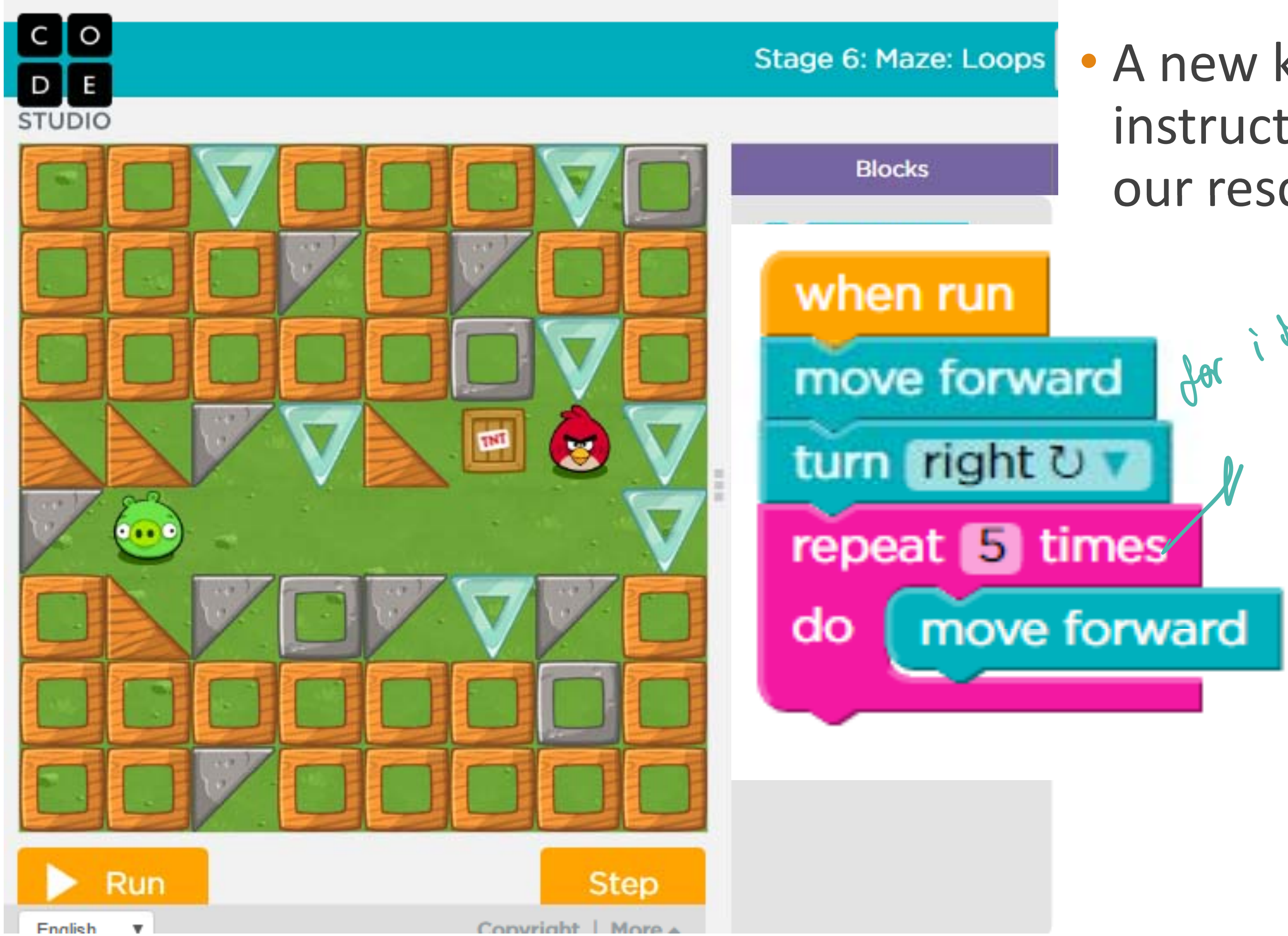


```
when run
move forward
turn right ↻
move forward
move forward
move forward
move forward
move forward
```

- How about this?
- Writing a straight-forward sequence of instructions would required a lot of instructions.

From: https://studio.code.org/s/course2/stage/6/puzzle/5 (CC-BY-NC-SA License)

A longer sequence (2)



```
when run
move forward
turn right ↻
repeat 5 times
do  move forward
```

for i in range(0,5)

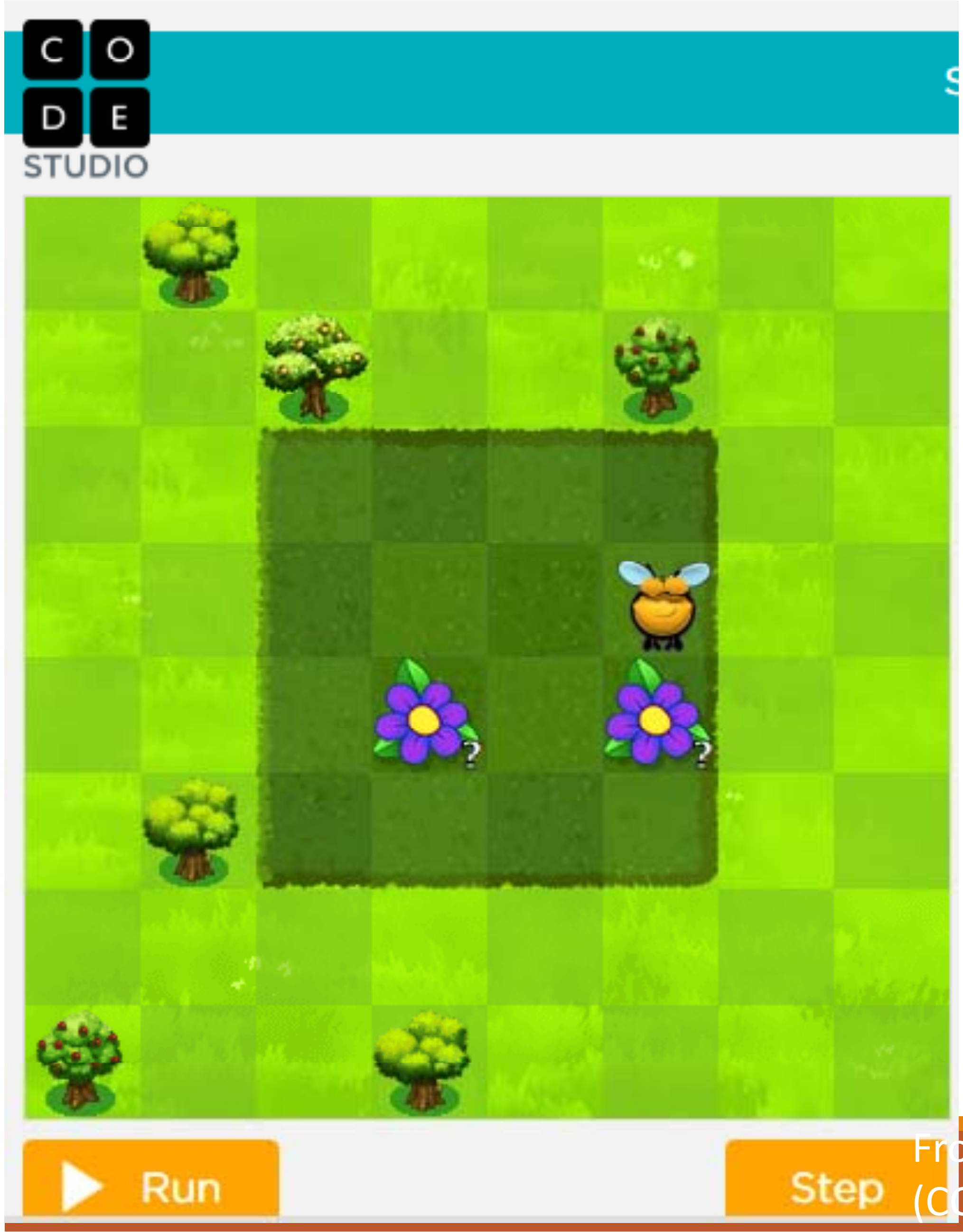
- A new kind of instructions comes as our rescue.

tell the bird to
thing
lly, instead of
g the
ons your self.

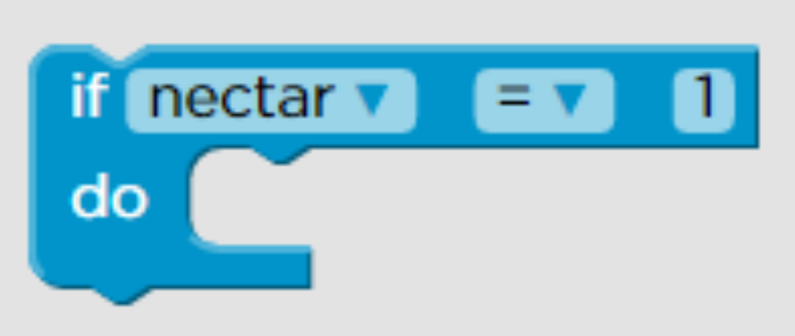
ould the
ons be now?

From: https://studio.code.org/s/course2/stage/6/puzzle/5 (CC-BY-NC-SA License)

A situation with uncertainty (1)



- You want to get all the nectar from both flowers, if possible, but you **do not know** if both flowers have nectar.
- The instructions to the bee should be flexible enough to handle all possibilities.

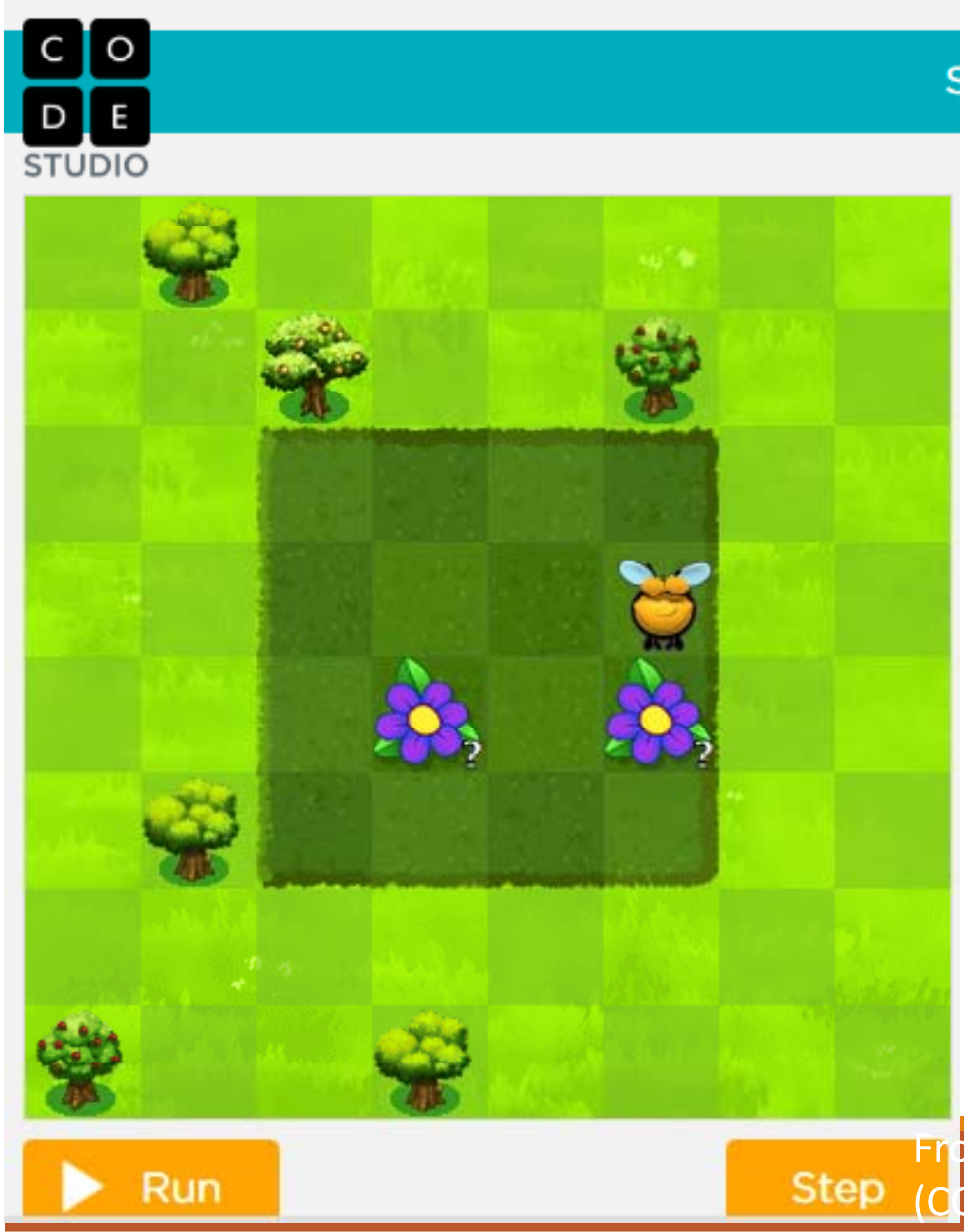


```
if nectar = 1
do
```

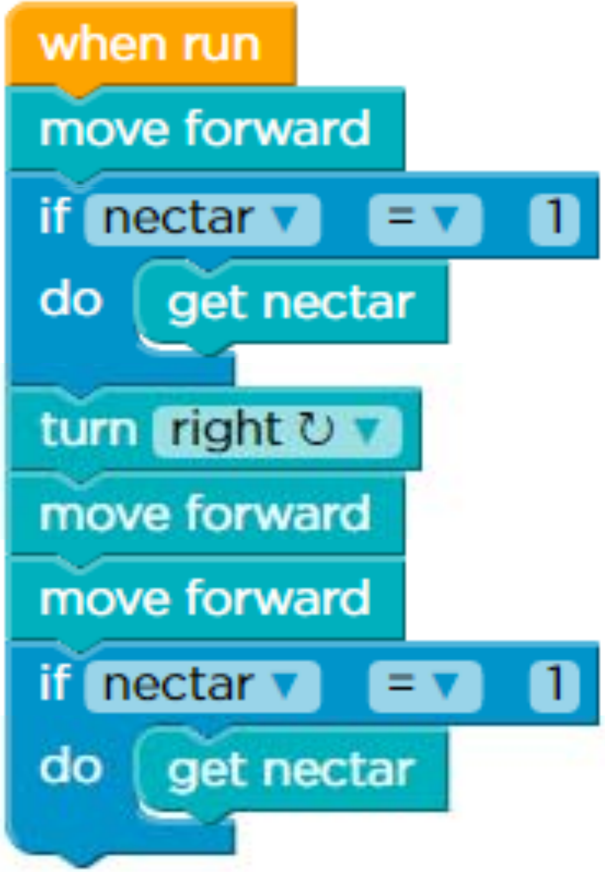
In most programming language, you can express conditions.

From: https://studio.code.org/s/course2/stage/13/puzzle/5 (CC-BY-NC-SA)

A situation with uncertainty (2)



- Using conditions, the code for this situation is:

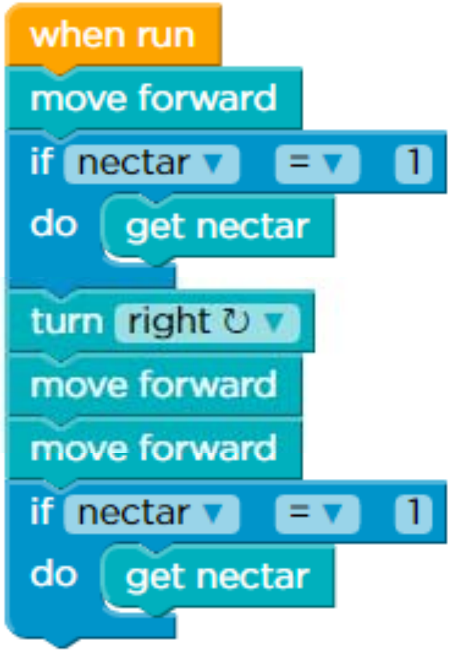


```
when run
move forward
if nectar = 1
do  get nectar
turn right ↻
move forward
move forward
if nectar = 1
do  get nectar
```

From: https://studio.code.org/s/course2/stage/13/puzzle/5 (CC-BY-NC-SA)

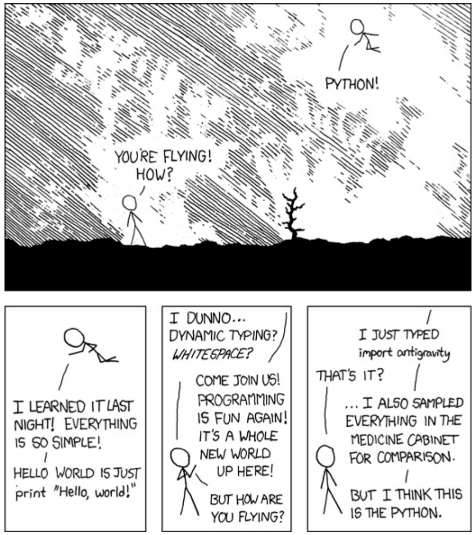
A real programming language

- While writing codes in blocks are fun, for longer programs, using this drag-and-drop approach is not very convenient.
- The previous bee code can be written in a typical programming language as follows.



```
when run
move forward
if nectar ▼ = ▼ 1
do get nectar
turn right ↻ ▼
move forward
move forward
if nectar ▼ = ▼ 1
do get nectar
```

```
MoveForward();
if(NectarRemaining() == 1) {
  GetNectar();
}
TurnRight();
MoveForward();
MoveForward();
if(NectarRemaining() == 1) {
  GetNectar();
}
```


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

ชั่วๆจริงเขียนอง

Building complex instructions with simple structures

- From previous code.org examples, you can see that for a given situation, there can be many ways to achieve your goal. Choosing the appropriate approach is fairly challenging.
- Loops and conditions are key tools we can use to express our ideas.
- In this course we will learn many other expressive structures that help you express your idea precisely and concisely.

The C# programming language

- In this course, we will use the C# programming language to teach you computer programming.
- Like natural languages (e.g., Thai, English, or Japanese), every programming language has its specific details and grammars.
- But we will focus more on learning how to program and how to write good codes, skills that can be applied with any popular computer programming languages.

C#

Why do Java progmmrs wear glases?
Becane they can't C#.

Let’s try C# (1)

- Guess what the following C# program does.

```
using System;

public class Program
{
  public static void Main()
  {
    Console.WriteLine("Hello C#");
  }
}
```

namespace

The output

```
Hello C#
```

While there are a lot of texts in the program, the main work-horse is the instruction: “Console.WriteLine(…)”, that displays “Hello C#” to the screen.

Let’s try C# (2)

- Guess what the following C# program does.

```
using System;

public class Program
{
  public static void Main()
  {
    int r;

    int a = int.Parse(Console.ReadLine());
    int b = int.Parse(Console.ReadLine());
    Console.WriteLine("{0} + {1} = {2}",a,b,a+b);
  }
}
```

The output

```
11
27
11 + 27 = 38
```

The program reads two integers, and outputs their summation.

That looks difficult…

```
int a = int.Parse(Console.ReadLine());
```

- If this is the first time you see computer programs, you may feel that this line of code may look fairly difficult.
- But as you continue to see and write more programs, it will be much easier to understand. You will learn more about C# syntax it in the next few weeks.

Let’s try C# (3)

- Guess what the following C# program does.

```
using System;
public class Program
{
  public static void Main()
  {
    double r;

    r = double.Parse(Console.ReadLine());
    Console.WriteLine("{0}",Math.PI * r * r);
  }
}
```

The output

```
10
314.15926538979
```

The program reads a number and outputs pi times that number squared….   What is the goal of this program, again?

It actually computes the area of the circle with radius **r**.

A better program

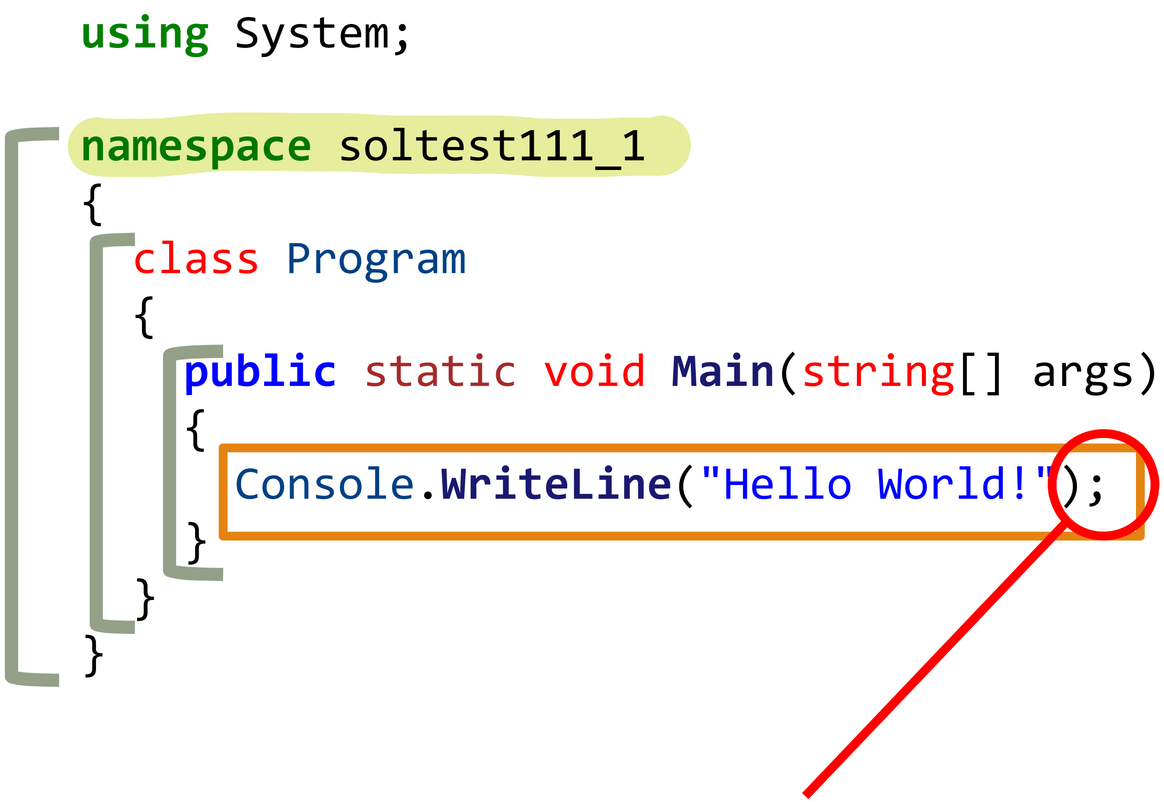
- The following fragment of the code, while performing the exact same task, is easier to understand, because it states its intention fairly clearly.

```
public static void Main()
{
  double r, area;

  r = ReadRadius();
  area = CircleArea(r);
  Console.WriteLine(”{0}”, area);
}
```

Basic C# program components

- A typical C# program looks like this one on the right hand side.
- In many cases, you will write instructions in the section outlined in the orange box.
- The other parts of your program declares other structures of you code.
  - You can see that they form nested structure of code using { }.

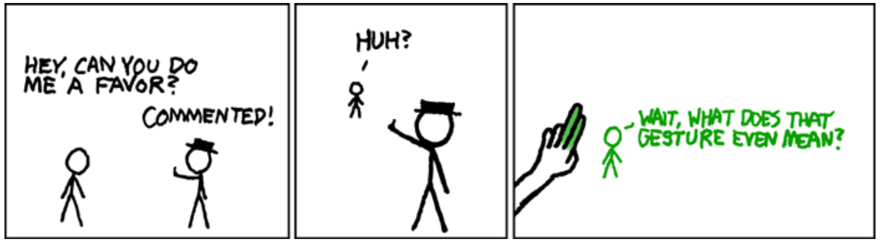


```
using System;

namespace soltest111_1
{
  class Program
  {
    public static void Main(string[] args)
    {
      Console.WriteLine("Hello World!");
    }
  }
}
```

You will also see a lot of semicolon (;).
This is how a statement in C# ends.

Comments


HEY, CAN YOU DO ME A FAVOR?
COMMENTED!
HUH?
WAIT, WHAT DOES THAT GESTURE EVEN MEAN?

Note: Your IDE colour may vary.

```
/*
 * Created by SharpDevelop.
 * Date: 8/7/2016
 * Time: 2:46 PM
 * To change this template use Tools | Options | Coding | Edit Standard Headers.
 */
using System;
namespace lab0001
{
  // This is also a comment, but it is a one-line comment.
  class Program
  {
     public static void Main(string[] args)
     {
        Console.WriteLine("Hello World!")

        // TODO: Implement Functionality Here

        Console.Write("Press any key to continue . . . ");
        Console.ReadKey(true);
     }
  }
}
```

- There are parts of your program that are intended for human to read. The compiler will ignore them.
- They are called code **comments**.
- They usually provide insights into how to code works or give notes.

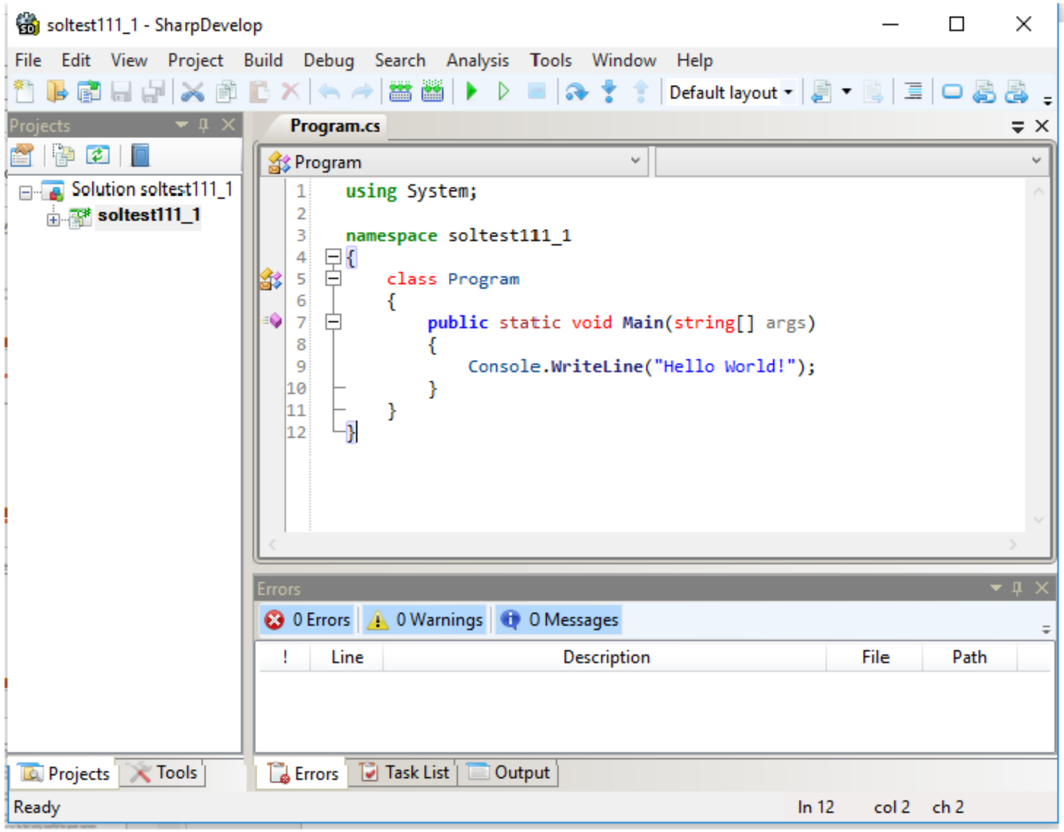
A lot of keywords

A sleepy class..

```
using System;
namespace soltest111_1
{
  class Program
  {
    public static void Main(string[] args)
    {
      Console.WriteLine("Hello World!");
    }
  }
}
```

- In order to translate your program written in C# into an executable code, the compiler has to know **exact** meaning of your code.
- The C# language uses many **keywords** to let you describe your ideas precisely. You will learn the usage of some of the keywords, but not all.
- We might not teach you the keywords that are not very crucial, so you need to just memorize and use them as provided by our examples.

IDE – where you write your C# programs



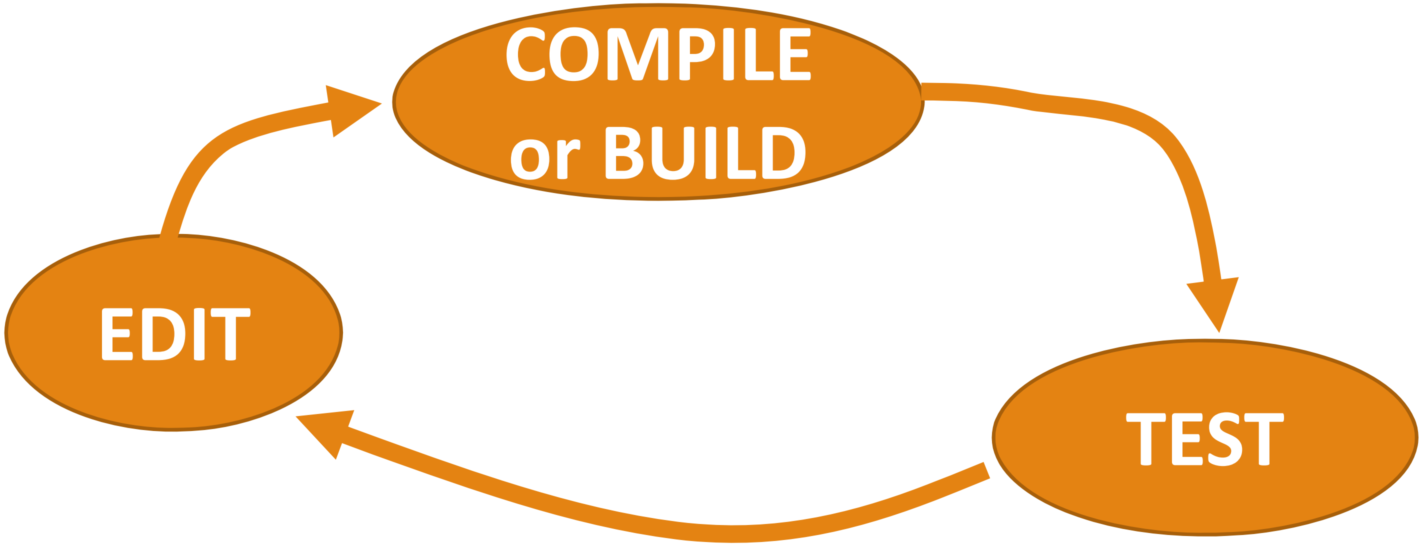
```
using System;

namespace soltest111_1
{
    class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

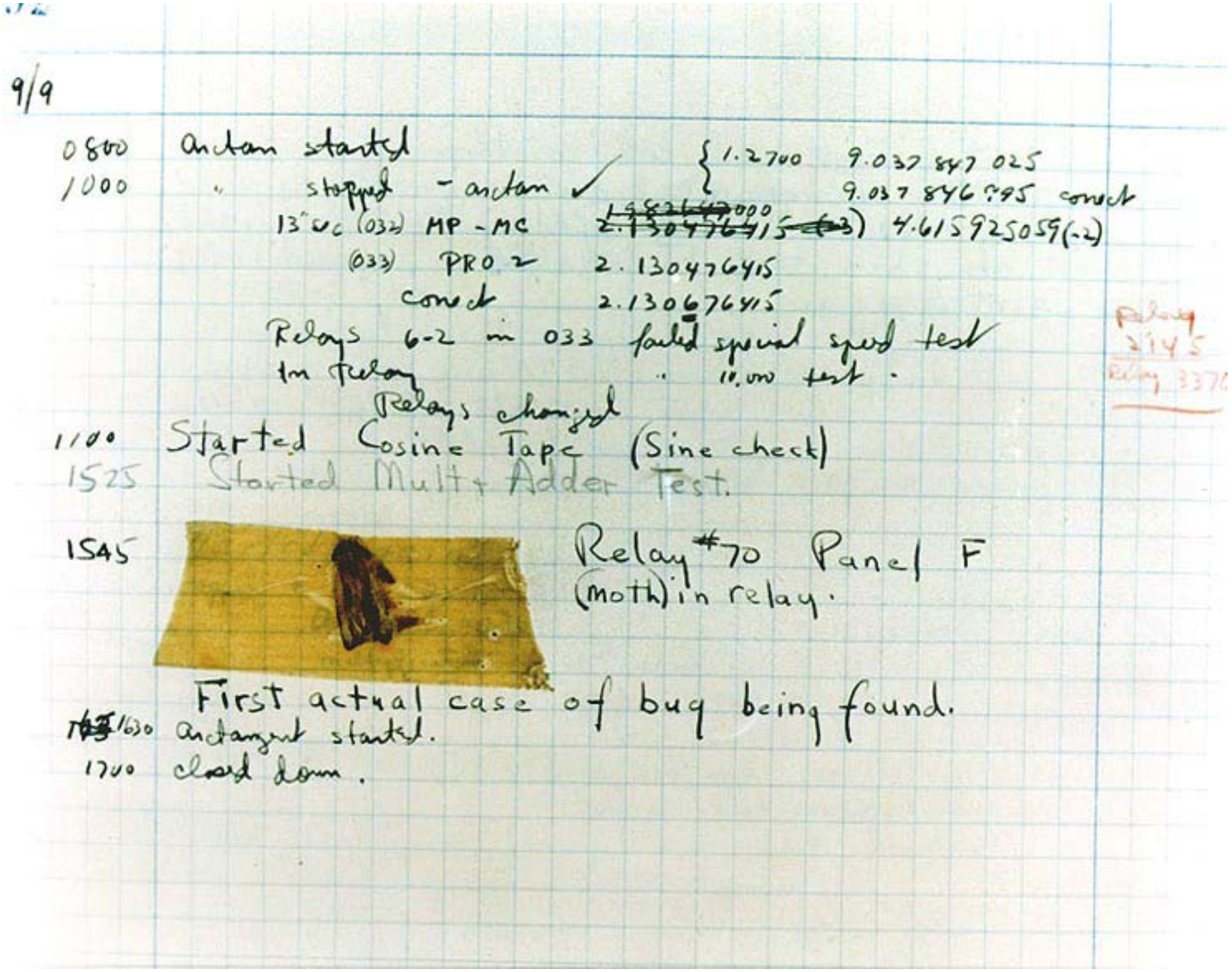
- We will write our C# programs in an application software that provides editing facility and compiling services for C#. This type of software is known as IDE’s (integrated development environment).

Edit-Compile-Test Loop

- Because everything in life doesn’t always work the first time you try it, your program may not always do exactly like what you want.
- It may be correct in some case, but it might fail in some other. Therefore, you need to **test** your program. If it is not correct, you have to **fix** it (**debug** it), and try to test it again.
- Your process for writing C# program would look like this.



Debugging



9/9

0800 Antan started

1000 " stopped - antan ✓ {1.2700 9.037 847 025
9.037 846 995 correct

13" uc (032) MP - MC ~~1.982147000~~ ~~2.130476415~~ (-3)) 4.615925059(-2)

(033) PRO 2 2.130476415

correct 2.130676415

Relays 6-2 in 033 failed special speed test

In relay " 11.000 test .

Relays changed

Relay 2145

Relay 3370

1100 Started Cosine Tape (Sine check)

1525 Started Mult+ Adder Test.

1545 Relay #70 Panel F (moth) in relay.

First actual case of bug being found.

1630 antangent started.

1700 closed down.

- When programmers try to fix mistakes in their codes, they usually refer to the activity as “debugging” (killing the bugs).
- In the early days of computing, there was actually a **bug** in the machine. Therefore, problems with computers are often referred to as **bugs**.

Conclusions

- Computer programming skills (or coding skills) are very important. That’s why you should learn them.
- A computer takes instructions in a machine readable form. We usually write codes in a higher level language. Before a computer can execute our instructions, they have to be translate into a form that it can understand.
- Computer programs consist of instructions. Many of them are control instructions. You can express complex ideas using these structures.
- Finally, you will learn to program using the C# programming language. At the end of the course, we hope that you would enjoy coding and find the course to be very useful to your career.

References

- You can look at computing history at
  - https://en.wikipedia.org/wiki/History_of_computing
  - http://www.computerhistory.org/timeline/computers/
- Learn how to program at code.org
  - http://www.code.org
- There are a lot of additional C# tutorials that you can read on-line
  - C# programming guide: https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx
  - http://www.tutorialspoint.com/csharp/
  - http://csharp.net-tutorials.com/