

# HIGH LEVEL LANGUAGES

Compilers  
Translators

# HIGH LEVEL LANGUAGES

- ◉ To run a program in a computer the program must first be loaded into **RAM**. Think of when you want to use a word processor, you firstly load the word processor into RAM. The processor of the computer carries out the instructions of the program in turn. Each instruction is fetched from main memory (RAM), decoded and then executed. This process continues until all instructions have been processed.
- ◉ Programs are written in High Level Languages (HLL) eg Visual Basic. The four features of HLL's are:-
  - They are written in **English**, errors are easier to find and fix
  - They must be **translated** into machine code (computers don't understand HLLs)
  - They are **designed to solve problems**
  - They are **portable** (can be used on different computer systems without alteration)

# HIGH LEVEL LANGUAGES

## 1. Instructions in English -

programs easy to understand, instructions are in everyday language. The instruction:

picDisplay.Print area   Prints **area** into pic box picDisplay

## 2. Must be Translated

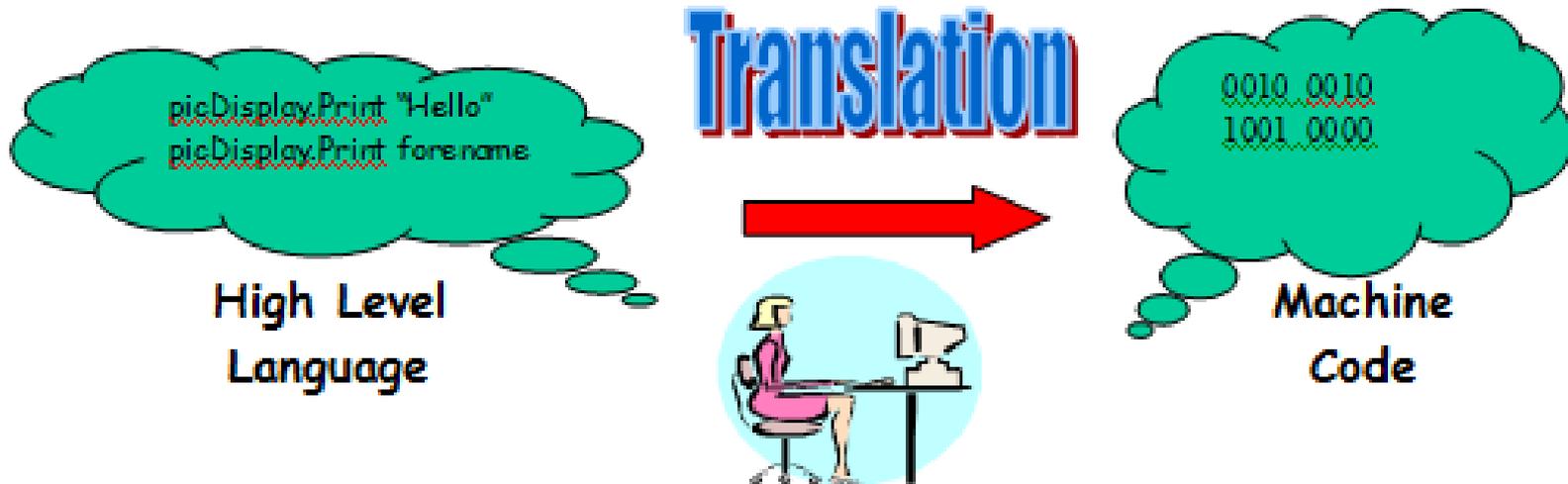
Each High Level Language instruction can be understood by us, but the computer only understands machine code (made up of 1s and 0s and sometimes called 'the computers own language'). Each HLL instruction must be translated into a number of machine code instructions.

# CONT.....

3. Designed to solve problems - programs are written to solve problems. Eg someone had the idea that it would be good to be able to find your car in a busy car park on your return to the car park. From this idea someone wrote a program to direct the driver from their current location to the location of their car using a GPS signal. So software has developed due to ideas beginning "wouldn't it be good if....." Can you think of any app you would want on your mobile?
4. Portable - software can be written to work on a number of different platforms, this means that irrespective of the hardware and software you have the program will still run. This means software companies can sell their software to more people, thereby increasing their profit.

# TRANSLATORS

- Changing a program from one computer language to another is called Translation.



- There are two different types of translators  
- **Interpreters** and **Compilers**

# COMPILER

- ◉ A **compiler** is a program. It takes an entire HLL program and translates it into Machine Code in a **single operation**.
- ◉ The original HLL program is called the **Source Code**.
- ◉ The Machine Code program produced by the compiler is called the **Object Code**. The compiler changes each HLL instruction into several Machine Code instructions.
- ◉ Object Code runs very fast as it is in the computer's own language.
- ◉ It's a good idea to save both the Source Code and the Object Code. The Object Code is saved as it is the version which is run. Can you think why the Source Code is saved?
- ◉ It is saved in case the original program gets edited in the future.

# COMPILER

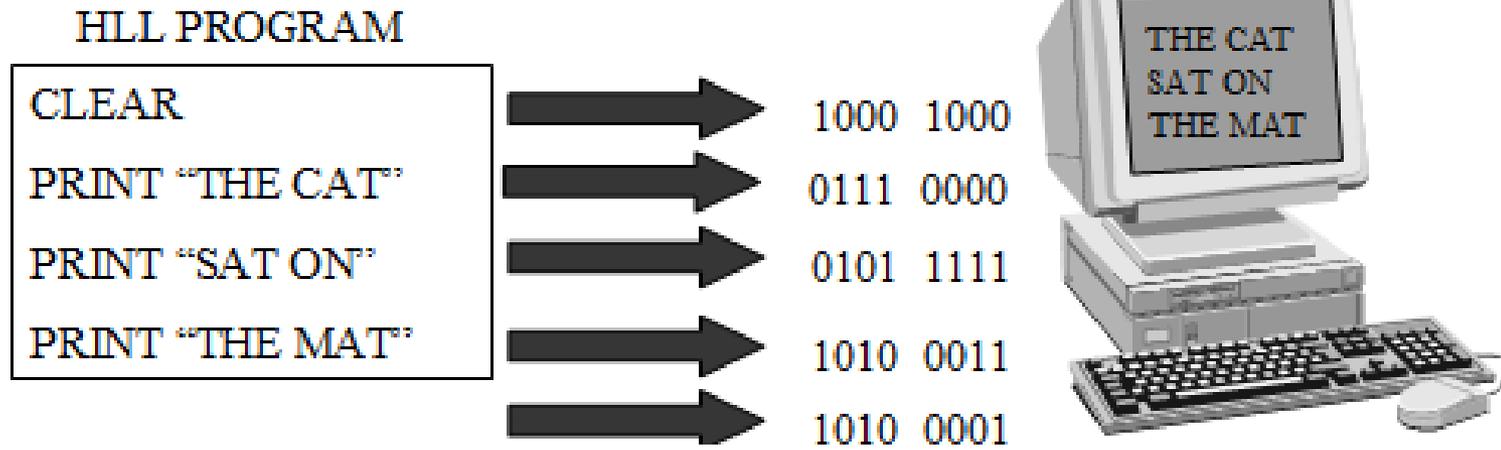
- ◉ The Source Code is translated, in a single operation, into Object code:



- ◉ The object code is not executed immediately, but can be saved separately.

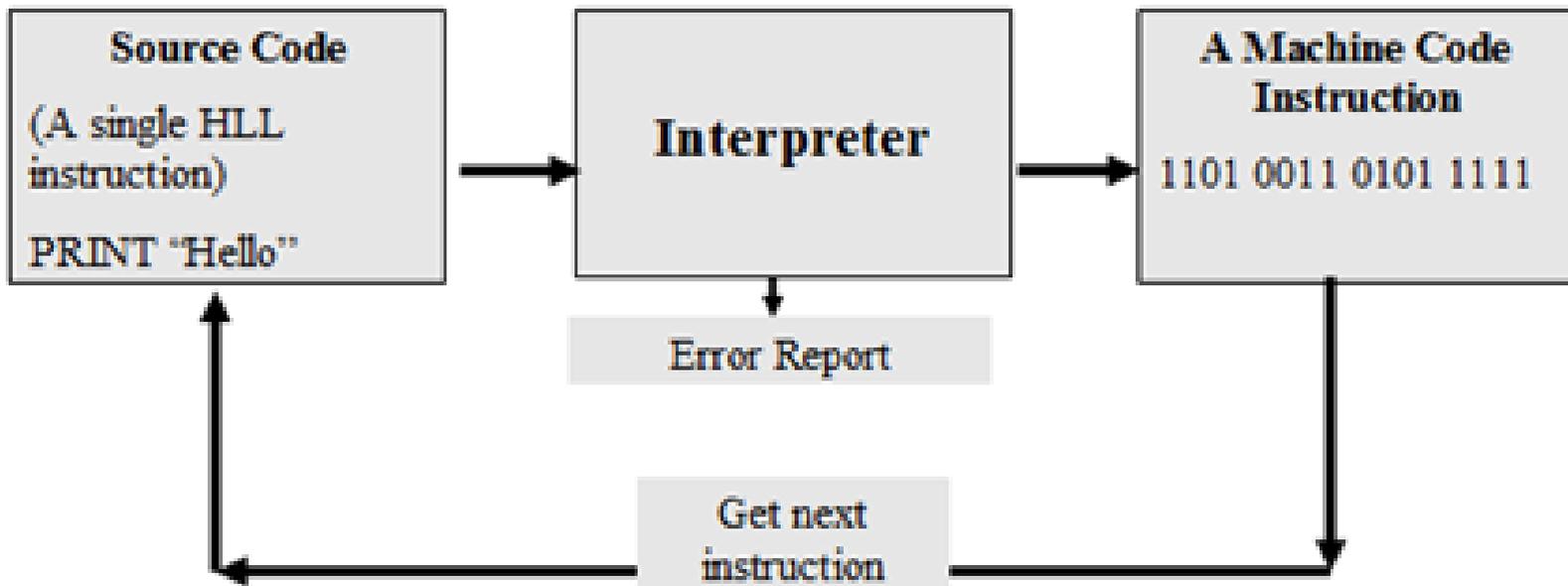
# INTERPRETER

- ◉ An **interpreter** takes a program written in High Level Language and translates it into Machine Code **one line at a time**. After each line is translated the command is carried out.



# INTERPRETER

- ◉ Interpreted programs run much more slowly than compiled programs. This is because the translator must translate each instruction from HLL into machine code, one instruction at a time. This must occur for each instruction every time the program is run. There is NO object code produced by an interpreter.



# COMPARISON

	<b>Compiler</b>	<b>Interpreter</b>
<b>Run Time</b>	Run time is fast as the code is already translated. The object code program is run so it does not need to be translated as it's already in machine code.	Run time slower, interpreter must translate each instruction in turn, every time the program is run. Code inside a loop must be translated and executed the number of times the loop repeats. For a loop for 10 each instruction inside the loop needs translated and executed 10 times.
<b>Ease of fixing errors</b>	Program will not compile if it has errors. Error report is produced after it tries to compile the code. The errors must be found and fixed in the HLL code and then re-compiled. More errors could result from this.	The program will run successfully up until the error is encountered. This allows partial code to be tested and allows the programmer to see the program improve. Syntax errors are highlighted immediately allowing them to be fixed.

# COMPARISON CONT...

	Compiler	Interpreter
<b>Use of RAM</b>	The compiler software can be removed from RAM once the code is compiled, this saves RAM in the future as only the object code program is needed in RAM to run the program. The resultant object code can be larger as RAM is not being used up storing the compiler software.	The interpreter software can never be removed from RAM, it must be present every time the program is run.  This means the resultant HLL code must be smaller as it cannot use the full amount of RAM.
<b>When should I use each?</b>	Use compiler for the final version of a program as it contains no errors and run time is fast.	Use when developing a program as it highlights error immediately making them easier to find and fix, and allows partial code to be run so you can see the result of coding up to that point in the program.

Complete Exercise 1 and get your teacher to check your answers.