

# **Computer Science and an introduction to Pascal**

A Level Computing – Mr Sheehan (MSH)

# Logging in

---

- ▶ **Username:** ict001-ict030
- ▶ **Password:** password

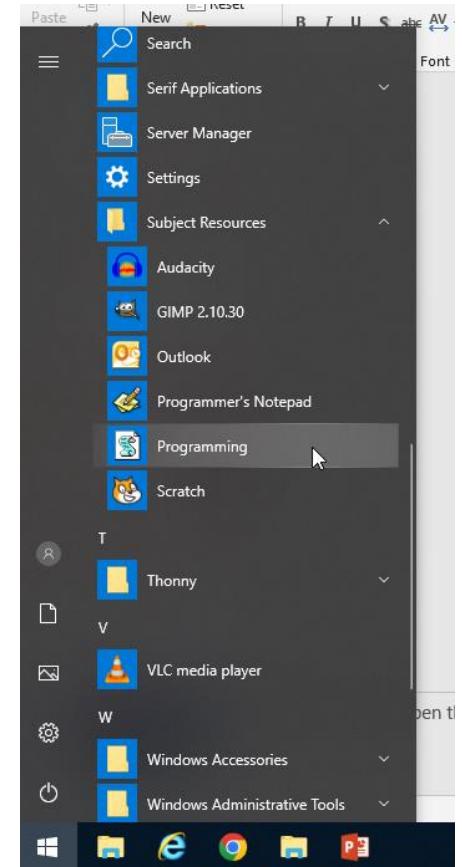


# Virtual Machine:

---

Start -> Subject Resources -> Programming

- ▶ What is a virtual machine?
- ▶ Why do you think we use a virtual machine for programming in school?



# IDE and Programming Language

---

The programming language we will be using in class is **Pascal**.

Most of you would have used Python for GCSE and so this will give you the experience of another language.

The IDE we currently use in lessons is **Lazarus** version 2.0.12  
This can be downloaded for free here:

<https://www.lazarus-ide.org/index.php?page=downloads>

► What is an IDE?



# Key words and terms

Term	Explanation
Statement	
Identifier	
Reserved words	
Subroutines	
Procedure	
Function	

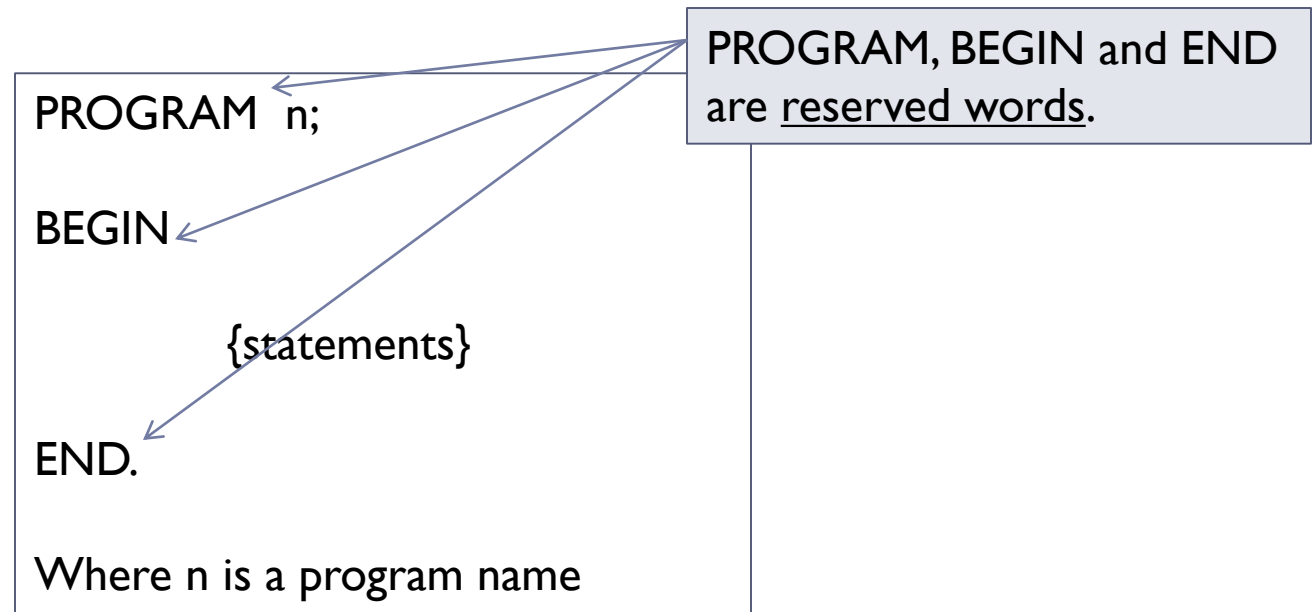
With the person next to you, discuss the definitions of these words in relation to programming.



# The Basic Form

---

- ▶ At its simplest, a Pascal program has the form



- ▶ Identifiers can only contain letters, underscores or numbers and must start with a letter.
-

# Getting Started

---

- ▶ Lazarus opens automatically when opening the virtual machine. Ensure you have it on screen.

## Click Start -> New -> Program

- ▶ The first line of the program is called the **program heading**. The rest of the program is the **program body**.
- ▶ At least one space or end-of-line must appear between adjacent identifiers, reserved words and numbers.



# First Program: Hello World!

---

- ▶ To see the effect of a program there must be some output. A single line of output can be produced using the writeln statement.

```
PROGRAM helloworld;  
  
BEGIN  
  
    writeln('Hello World!');  
  
END.
```

- ▶ The text to be output must always appear within apostrophes or string quotes.
  - ▶ Statements must always end with a semi-colon.
  - ▶ Copy this code. Run the program. What is the problem with it?
  - ▶ Add **readln()** under **writeln('Hello World!');** to fix this.
- 





# What is wrong with the code?

---

```
PROGRAM while;  
  
BEGIN  
  
    writeln('Hello World!');  
  
END.
```

```
PROGRAM helloworld;  
  
BEGIN  
  
    writeln('Hello World!')  
  
END.
```

```
PROGRAM helloworld;  
  
BEGIN  
  
    writeln(Hello World!);  
  
END.
```

```
PROGRAM 2_helloworld;  
  
BEGIN  
  
    writeln('Hello World!');  
  
END.
```



# Try the following Output Examples

```
PROGRAM helloworld;  
  
BEGIN  
    writeln('Hello', 'World!');  
    readln();  
  
END.
```

```
HelloWorld!
```

```
PROGRAM helloworld;  
  
BEGIN  
    writeln('Hello');  
    Writeln('World!');  
    readln();  
  
END.
```

```
Hello  
World!
```

```
PROGRAM helloworld;  
  
BEGIN  
    writeln('Hello');  
    writeln();  
    writeln('World!');  
    readln();  
  
END.
```

```
Hello  
World!  
-
```

```
PROGRAM helloworld;  
  
BEGIN  
    write('Hello');  
    writeln('World!');  
    readln();  
  
END.
```

```
HelloWorld!
```

# Output

---

- ▶ Spaces and end-of-line make no difference to the compiler. To the human reader, however, their inclusion is very important; neat layout is a *major* contributory factor towards program transparency.
- ▶ Comments are helpful to identify what is happening in a particular part of the program, not only for you but anyone else who reads the code.

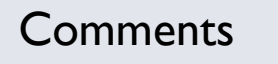
```
PROGRAM helloworld;
```

```
BEGIN
```

```
//Prints "Hello World! To the screen  
{Prints "Hello World! To the screen}  
writeln('Hello World!');
```

```
END.
```

Comments



# Exercises

---

1. Print your name to the screen.
2. Print your name to the screen in a border made of a character of your choice.
3. Write a program to draw a rocket.

```
      *
     ***
    *****
   *****
  *****
 *****
*****
*****
*****
```

4. Write a program to print your initials in large letters.

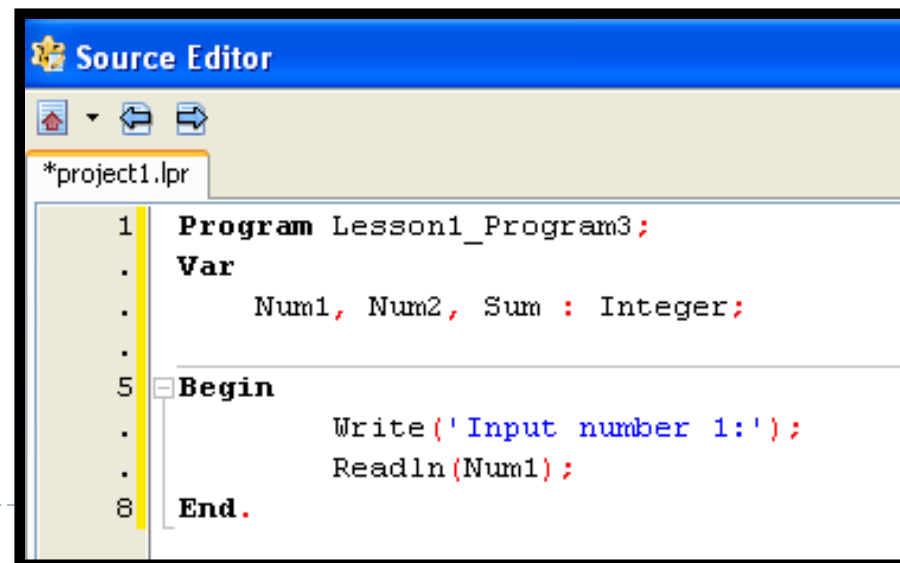
```
MM      MM      SSSSS
MMM     MMM     SS
MM MM MM      SSSSS
MM      MM      SS
MM      MM      SSSSS
```



# Extension

The following code has 3 integers (whole numbers) set up to be used in the program. Currently the program asks the user for a number and stores it in the variable Num1

Complete the program so the user must enter 2 numbers. Then have the sum of those two numbers printed to the console.



```
Source Editor
*project1.lpr
1  Program Lesson1_Program3;
.  Var
.    Num1, Num2, Sum : Integer;
.
5  Begin
.    Write('Input number 1:');
.    Readln(Num1);
8  End.
```