O/L 2023 ICT

SEMINAR

CONTENTS

PRASANNA SILVA

B.SC. IN IT

Grade 10 – Unit 7

SPREADSHEET SOFTWARE

Summary

An electronic spreadsheet is a software application that allows for calculations and the arrangement of data in rows and columns, much like a square-ruled book.

Examples:

Proprietary	Open Source	Online
• Microsoft Excel • Apple Numbers	• LibreOffice Calc • OpenOffice Calc	• Google Sheets • MS Office 365 Excel

UI Components

Active Cell	The cell currently selected.
Name Box	Shows the address of the selected cell.
Formula Bar	Allows entering, editing, and copying formulas.
fx Button	A wizard for easy function insertion.

Core Parts of Spreadsheet Software

Worksheet	Grid of columns and rows creating cells, identified by sheet tabs.
Workbook	Contains one or more worksheets, saved as a file (.xls, .xlsx, .ods).
Columns/Rows	Identified by letters and numbers respectively.
Cell	Intersection of a row and a column with a unique address (e.g., A1).
Cell Range	A block of adjacent cells selected together (e.g., B2:C5).

Data Types:

Labels	Values	Formulas
Text, left-aligned	Numerical, right-aligned	Begin with "=", outcome alignment varies

Formulas

Operator Precedence

Order	Operator	Operation
1	( )	Brackets
2	^	Power
3	* /	Mul / Div
4	+ -	Add / Sub

Functions

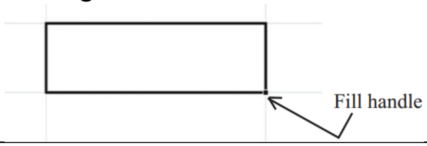
SUM()	Total of values
AVERAGE()	Average of the values
MIN()	Smallest value
MAX	Largest value
COUNT	Count of the numeric values

Cell Referencing

There are two types of cell referencing.

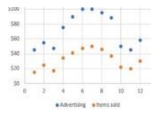
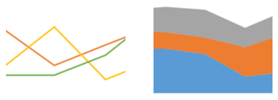
1. **Relative Cell Referencing** - Updates cell addresses based on the formula's copied location
2. **Absolute Cell Referencing** – Cell addresses remain constant, marked with "$".

Copying Formulas

Method 1	Method 2
• Select the cell • Drag the fill handle 	Select the cell with formula → Ctrl + C select the cells to be pasted → Ctrl + V

Charts

Purpose: Graphically represent data for easier understanding.

Bar/Column	Pie	XY Scatter	Line / Area
 For multiple columns and rows	 For percentages	 To observe trends	 To indicate change with time

Grade 11 – Unit 5

Web Design using HTML

- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- HTML is not a programming language, it is a **markup language**
- HTML uses **markup tags** to describe web pages

HTML Tags

- HTML tags are keywords surrounded by **angle brackets** like <html>
- HTML tags **come in pairs** like <b> and </b>
- Some tags do not have a pair. They are called "**Non container tags**" Eg: <hr>, <img>

HTML Attributes

- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes come in name/value pairs like: **name="value"**

Parts of an HTML Document

1. Head Section

contains information about the current document, such as its **title**.

2. Body Section

Contains the document's content such as text, images, animations, links, tables, frames etc.

Creating an HTML Document.

You can use following tools to develop web pages.

1. **Text Editor** (Notepad, Notepad++)
2. **Web Authoring Tool** (Dreamweaver, Bluefish)
3. **WYSIWYG Editors** (Muse)
4. **CMS** (Joomla!, Moodle, php-fusion)

Comments

Can be used to explain the source code.

Eg: `<!-- This is a comment Line -->`

Headings

There are 6 levels of headings in HTML.

```
<h1 align="center"> Largest </h1>
<h6 align="right"> Smallest </h6>
```

Paragraphs

A paragraph will have spaces before and after.

Eg: `<p align="right"> A paragraph </p>`

Line Breaks

breaks Use the <br> or <br/> tag if you want a line break (a new line) without starting a new paragraph:

Eg: `This is line 1 <br> This is line 2`

Font

Specify the font size, font face and color of text:

```
<font face="verdana" size="3" color="green">
This is some text!</font>
```

Text Formatting

Bold text	- <b> </b>	<strong></strong>
Italic Text	- <i> </i>	<em> </em>
Underlined	- <u> </u>	
^{Super}script	- <sup> </sup>	
$_{Sub}$script	- <sub> </sub>	
`Code text`	- <code> </code>	

HTML Escape characters

&amp; → &	&lt; → <	&gt; → >
&quot; → "	&apos; → '	&nbsp;→ space

Pre-formatted Text

Displays the text as it is with spaces and line breaks.

```
<pre>
Mr. Dasun Kariyapperuma
      No 123/B, Narahenpita,
</pre>
```

RGB Colors

Colours in HTML can be given in the following manner.

1. By name
2. Using the hexa-decimal value after #

```
Eg:    <body  bgcolor="red">
       <body bgcolor="#ff0000">
```

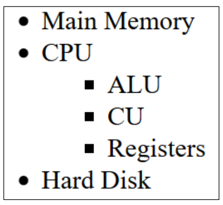
Links

$$\text{<a href= "}\underbrace{\text{page1.html}}_{\text{URL}}\text{" }\underbrace{\text{Target="\_blank"}}_{\text{Opens the link in a new tab}}\text{> }\underbrace{\text{Click here}}_{\text{Link text}}\text{ </a>}$$

Lists in HTML

1. Unordered List

```
<ul type="Disc">
   <li> Main Memory
   <li> CPU
   <ul type="square">
       <li> ALU
       <li> CU
       <li> Registers
   </ul>
   <li> Hard Disk
</ul>
```

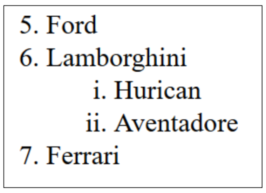


- Main Memory
- CPU
  - ALU
  - CU
  - Registers
- Hard Disk

types of lists : Disc , Circle , Square

2. Ordered Lists

```
<ol type="1" Start=5>
   <li> Ford
   <li> Lamborghini
   <ol type="i">
       <li> Hurican
       <li> Aventadore
   </ol>
   <li> Ferrari
</ol>
```



5. Ford
6. Lamborghini
   i. Hurican
   ii. Aventadore
7. Ferrari

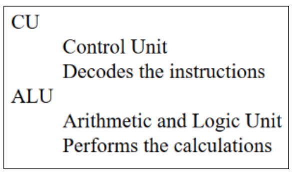
Different types of lists

1 – 1,2,3,4,5....
A – A,B,C,D...
a – a,b,c,d...
i – i,ii,iii...
I – I,II,III...

3. Description Lists

dt – Description Term
dd – Definition Details

```
<dl>
<dt>CU</dt>
  <dd>Control Unit</dd>
  <dd>Decodes the instructions</dd>
<dt>ALU</dt>
  <dd>Arithmetic and Logic Unit</dd>
  <dd>Performs the calculations</dd>
</dl>
```



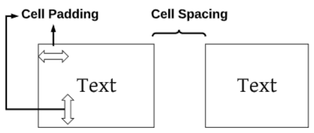
CU
    Control Unit
    Decodes the instructions
ALU
    Arithmetic and Logic Unit
    Performs the calculations

Tables

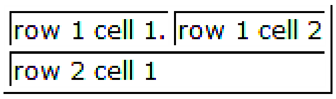
A table is divided into rows (with the <tr> tag), and each row is divided into data cells (<td>) and headings (<th>)**Colspan** and **rowspan** can be used to merge cells.

Table attributes

- **border** = default 0
- **cellpadding** = "5px"
- **cellspacing** = "0px"
- **width** = "400px" , **height** = "300px"

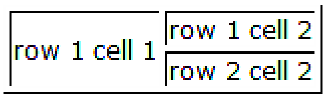


```
<table border=1>
<tr>
   <td> row 1 cell 1 </td>
   <td> row 1 cell 2 </td>
</tr>
<tr>
   <td colspan="2">row 2 cell 1</td>
</tr>
</table>
```



row 1 cell 1.	row 1 cell 2
row 2 cell 1	

```
<table border=1>
     <tr> <td rowspan="2">row 1 cell 1</td>
          <td>row 1 cell 2</td>
     </tr>
     <tr><td>row 2 cell 2</td></tr>
</table>
```



row 1 cell 1	row 1 cell 2
	row 2 cell 2

Caption

Shows a table title on the top of the table.

```
<caption>Table Title</caption>
```

The align=bottom attribute makes the caption to appear at the bottom of the table.

Images

$$\text{<img src="}\underbrace{\text{images/angry.gif}}_{\text{Image path}}\text{" alt="}\underbrace{\text{Angry}}_{\text{Alternate text}}\text{" />}$$

align	top, bottom, middle, left, right
border	*Pixels*
height	*Pixels, %*
usemap	*#mapname*
width	*Pixels, %*

Grade 11 – Unit 1

Programming - Pascal

Pascal is a **high-level, procedural, compiled** Programming language.

Identifiers

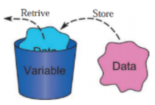
- Reserved words are not allowed
- Identifiers can only contain letters, numbers and the underscore.
- Identifiers should start with an English letter and should not contain spaces between words.

Data Types in Pascal

- **Integer** – positive or negative whole numbers.
- **Real** – Positive or negative decimal numbers.
- **Boolean** – True or false values
- **Char** – A single character
- **String** – A sequence of characters.

Variables

Variables store and allow manipulation of data in programming



Operators

The following basic types of operators are used in Pascal.

Arithmetic Operators

+	Addition
-	Subtraction
*	Multiplication
/	Division
DIV	Rounded Division
MOD	Modulus / Remainder

Operator Precedence

The execution order of operators

1. **NOT**
2. *** , / , DIV , MOD , AND**
3. **+ , - , OR**
4. **= , <> , < ,<= , > , >=**

Writing Pascal Programs

- **read()**, **readln()** - Input
- **write()**, **writeln()** - Output
- **":="** is the assignment operator.
- Each statement is terminated by a **semicolon (;)**
- **//**, **{ }** and **(* ….. *)** used for comments.
- **Uses crt;** allows to execute
  - **clrscr; readkey; gotoxy(x,y); textcolor(red);**
- Hold the output at the end using a **readln;**

Selections (Conditional statements)

IF Conditions

```
if N1 > N2 then
      Large := N1
else
      Large := N2;
```

Case Statements

```
Case Marks of
      0..34 : Grade := 'W';
      35..49 : Grade := 'S';
      50..64 : Grade := 'C';
      65..74 : Grade := 'B';
      75..100 : Grade := 'A';
Else
   Writeln('Invaid Marks');
End;
```

Repetitions

There are three types of loops in Pascal.

1. For – do

```
For count := 1 to 10 do
   writeln(count);

For count := 10 downto 1 do
   writeln(count);
```

2. While – do

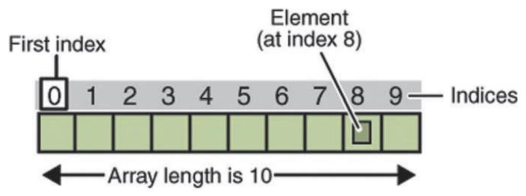
```
number := 1;
while number <= 10 do
   writeln(number);
      number := number + 1;
```

3. Repeat-until

```
count = 0;
Repeat
      writeln ('Pascal');
      count := count + 1
Until count > 5
```

Arrays

An array is a data structure that allows to store multiple items of the same data type using a single identifier name.



Defining an array

```
Var marks: array[0..9] of integer;
```

Assigning a value to an array element

```
marks[3]:=35;
```

Using a loop to access an array.

A for loop is used to traverse an array since it can iterate a specific number of times.

```
var ictm : array[1..40] of integer;
i,marks : integer;
for i := 1 to 40 do
    begin
          writeln('Enter marks');
          read(marks);
          ictm[i] := marks;
    end;
```

Sub programs

There are two types of subprograms

1. **Functions** – Returns a value.
2. **Procedures** – Does not return a value.

These subprograms are defined before the main program begins and is called from the main body of the program.

Functions

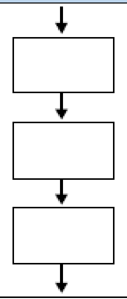
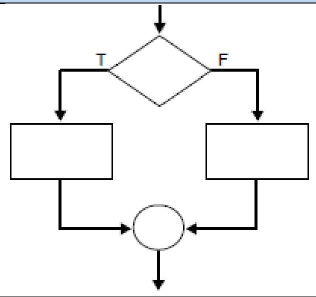
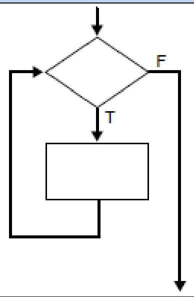
```
program exFunction;                    return type
var a, b, ret: integer;
function max(num1, num2:integer):integer;
var result: integer;
begin
   if (num1 > num2) then
      result := num1
   else
      result := num2;
   max := result;
end;
begin
   a := 100;                     function call
   b := 200;
   ret := max(a, b);
   writeln( 'Max value is : ', ret );
end.
```

Procedures

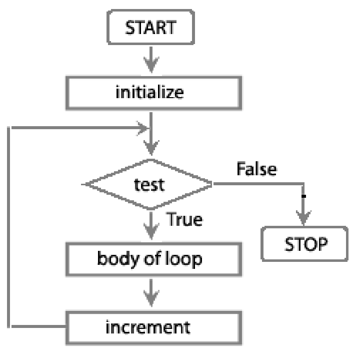
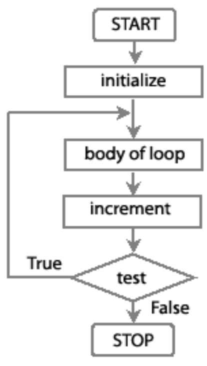
```
Program proctest;
Uses crt;
Var a,b,c,min:integer;
Procedure findmin(x,y,z:integer;
var m:integer);
begin
      if x<y then m:=x
      else m:=y;
      if z<m then m:=z;
end;
begin
      write('Enter three numbers: ');
      readln(a,b,c);
      findmin(a,b,c,min);
      writeln('Minimum number: ',min);
readkey;
end.
```

ALGORITHMS

- An algorithm is a step-by-step process of solving a problem or performing a task.
- Algorithms can be represented as **flowchart** and **pseudocode**.
- The flow of an algorithm can be controlled using **control structures**.

Sequence	**Selection**	**Loops / Repetitions**
		
• Each step is executed only once. • Flows from top to bottom.	• A control path is selected based on a question / condition	• Repeatedly executes statements based on a condition.

Loops / Repetitions

Pre-test Loops		Post-test Loops
While – end while Loop	For Loop	Repeat-Until Loop
 START initialize test False True STOP body of loop increment		 START initialize body of loop increment True test False STOP
• Executes while the condition is true. • Condition tested at the beginning of the loop • Exits the loop if condition is false at beginning.	• Iterates for a specific number of times. • Used to access elements in an Array.	• Iterates until a condition becomes true. • Check the condition at the end of the loop. • Loops at least once

`Begin`
`Initialize`
`While` `test` **`Do`**
`    body of loop`
`    Increment`
`EndWhile`
`End.`
	`For <var> = start`
	`TO/DOWNTO end`
	`    body of Loop`
	`End For`
		`Begin`
		`Initialize`
		`Repeat`
		`    body of loop`
		`    Increment`
		`Until` `test`
		`End.`