

Adam Mukharil Bachtiar


English Class

Informatics Engineering 2011

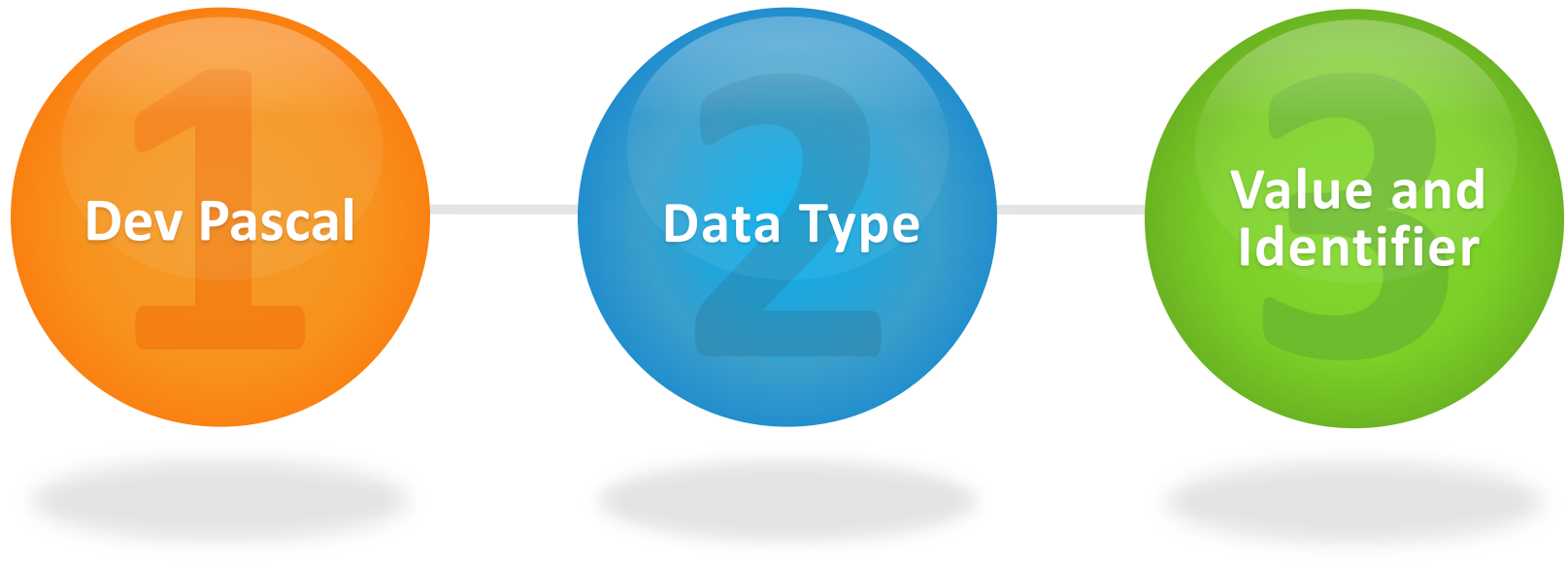


Algorithms and Programming

Introduction of Dev Pascal, Data Type, Value, and Identifier



Steps of the Day



Let's Start 





Dev Pascal

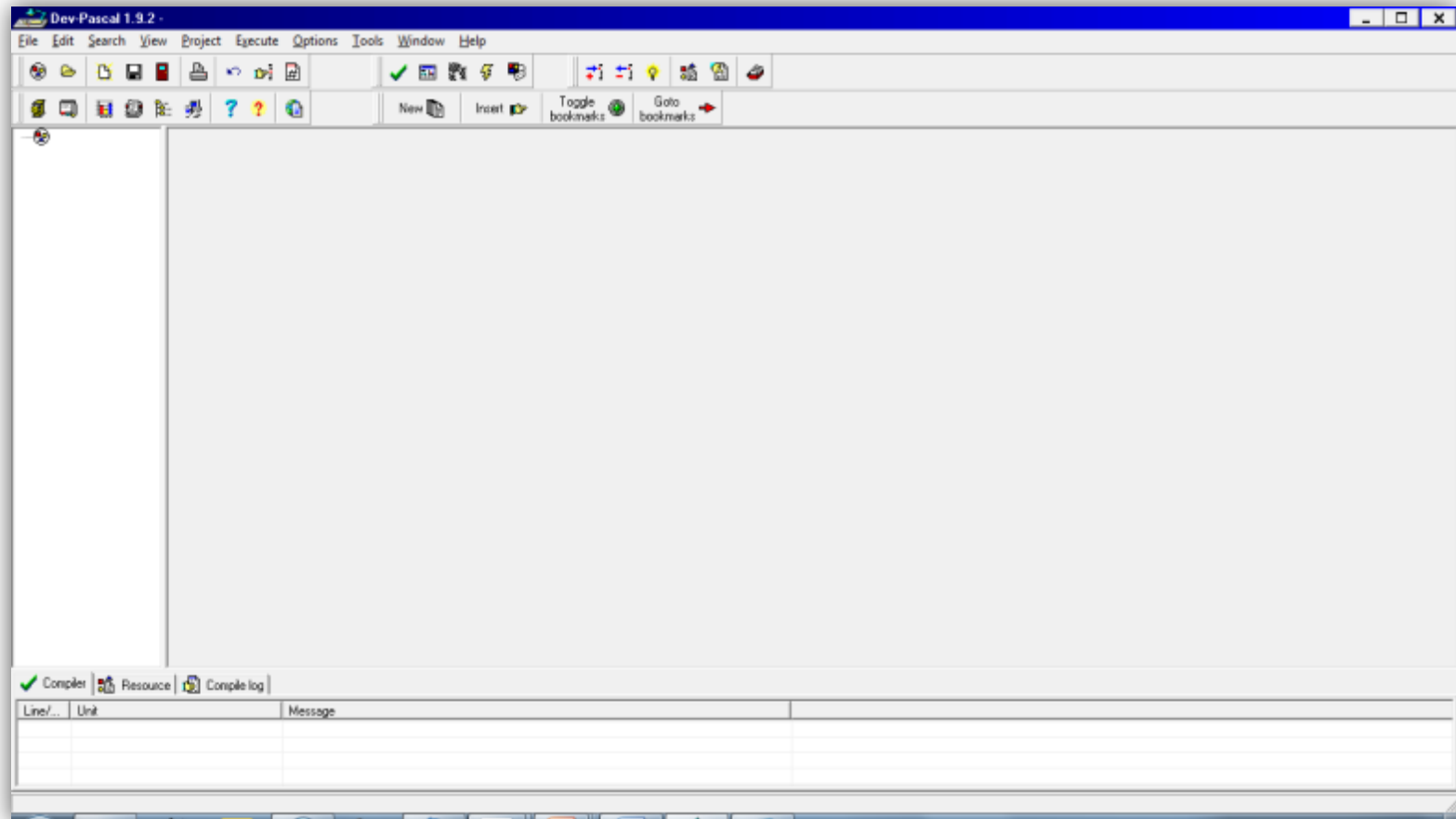
Definition and Instalation

An **IDE** (Integrated Development Environment) for **PASCAL** language that was built by **BLOODSHEED**. It's **Freeware**.

How to Make a Program in DEV Pascal?

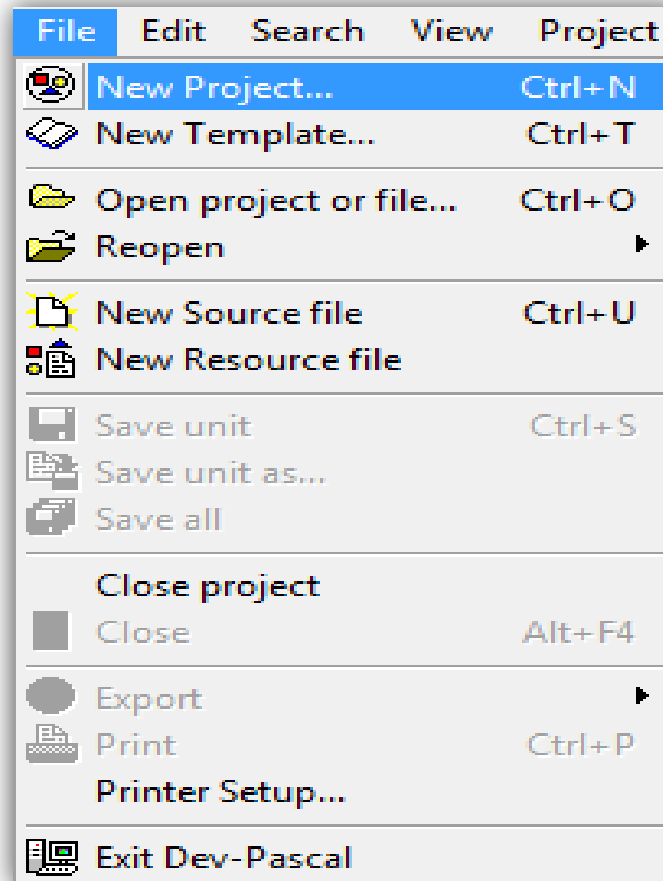


Step 1



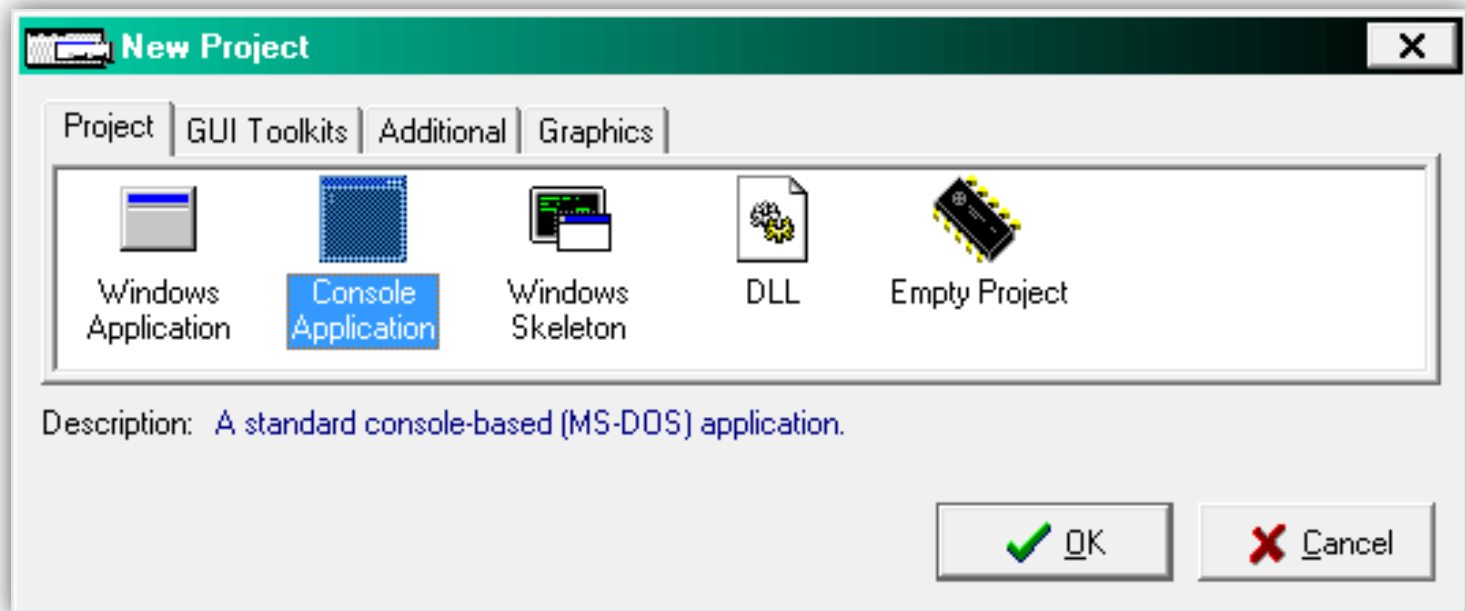
Open **Dev Pascal** application

Step 2



Make a **New File** or **New Project**

Step 3

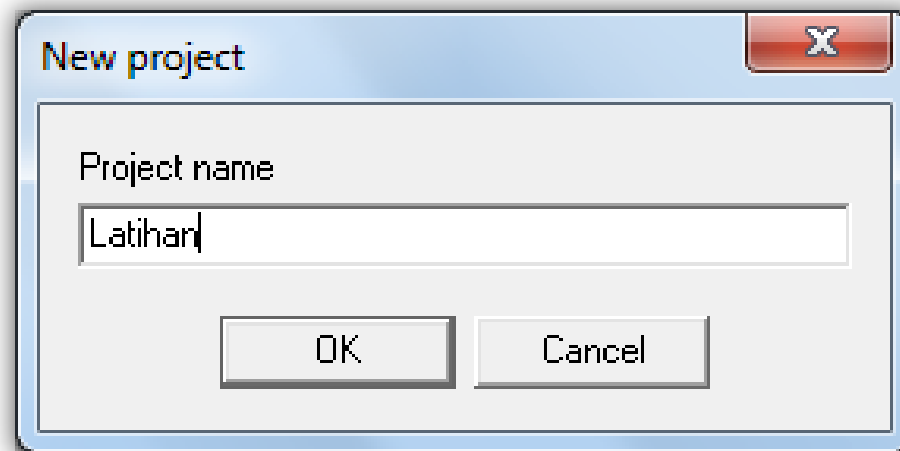


Choose **Console Application** → **OK**

Step 4

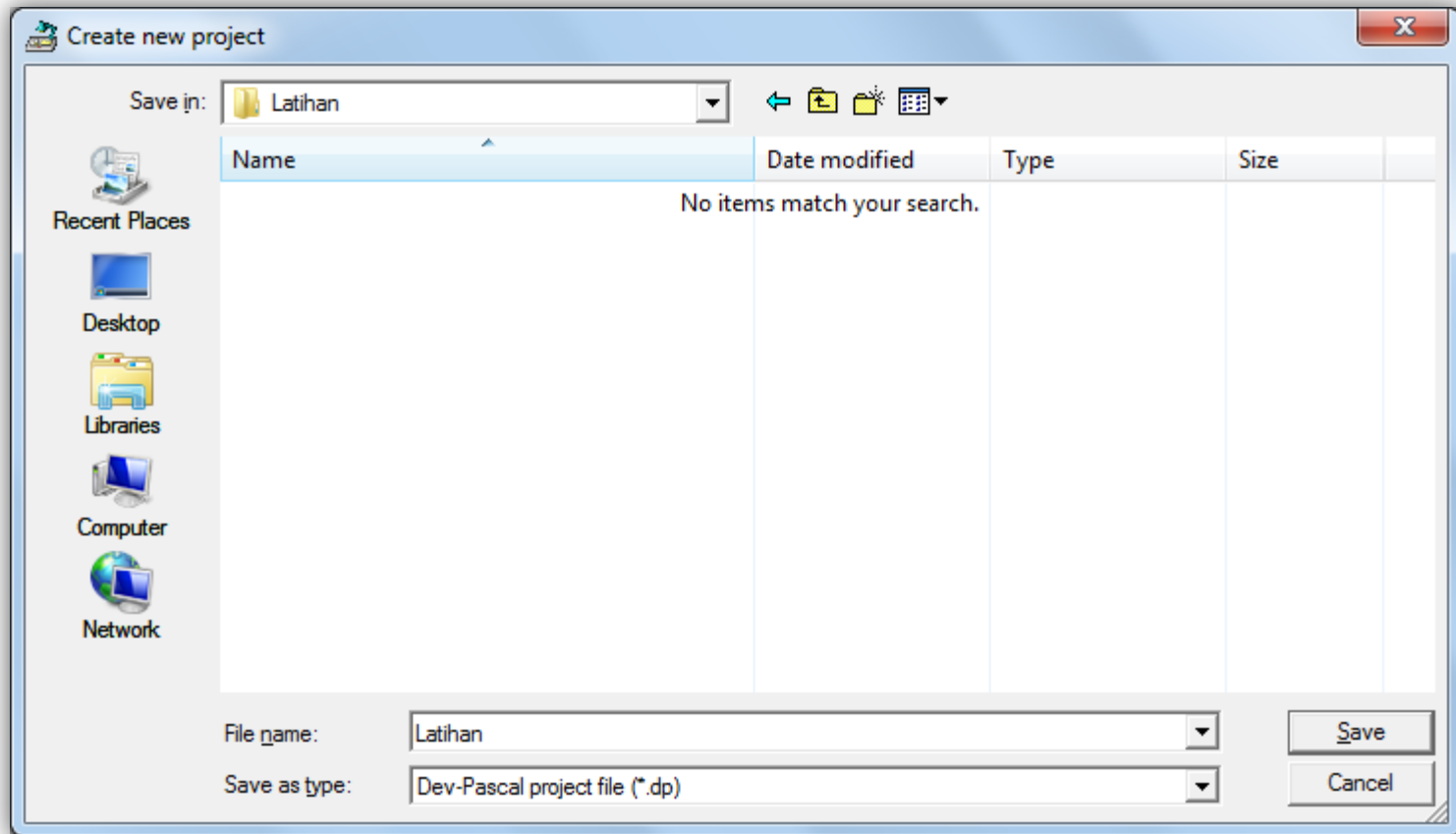
Give a name to project

(Name can contain space character)



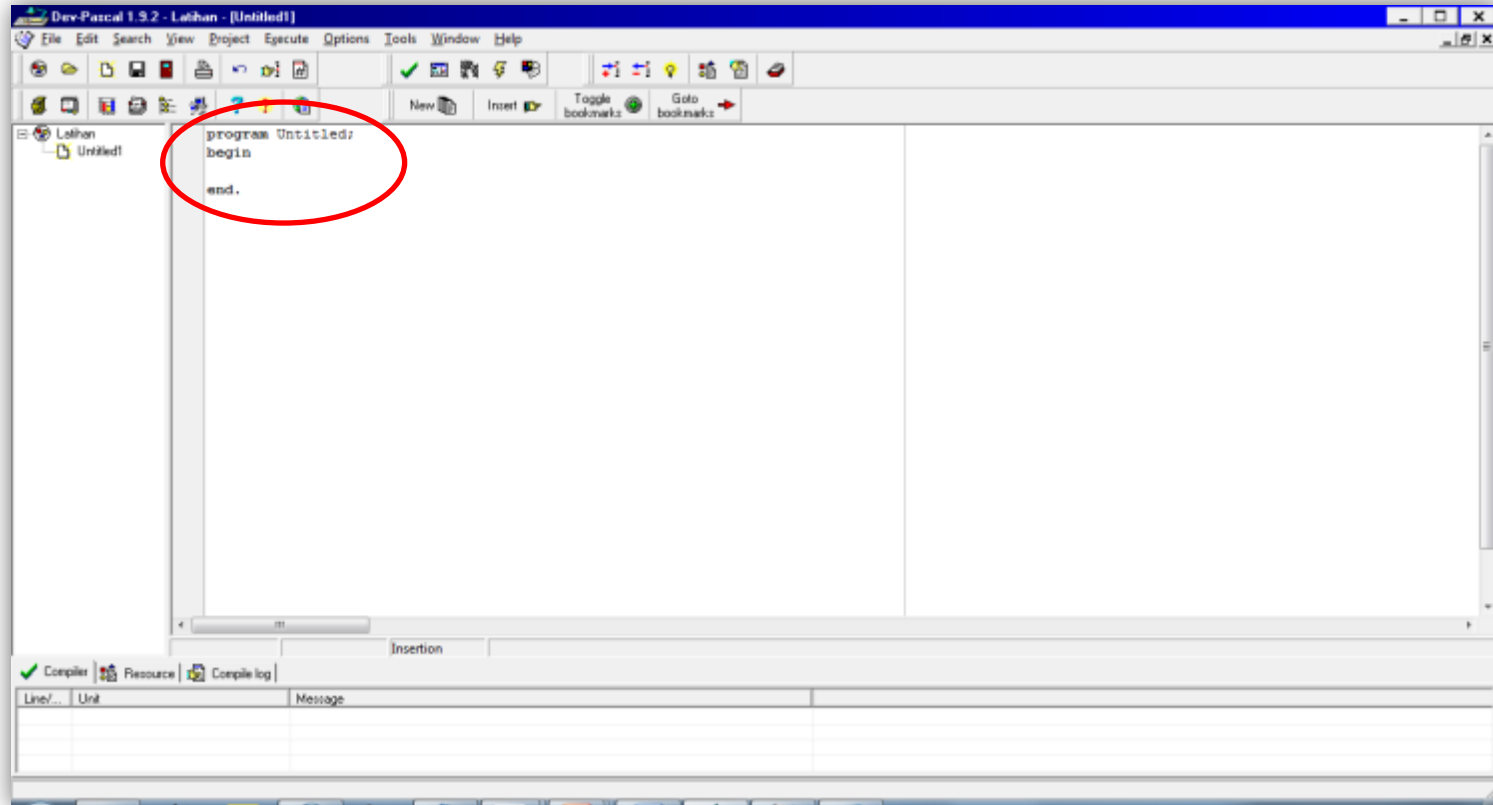
WARNING: Name of project should be same with name of its folder. One folder is only for one project (in my class)

Step 5



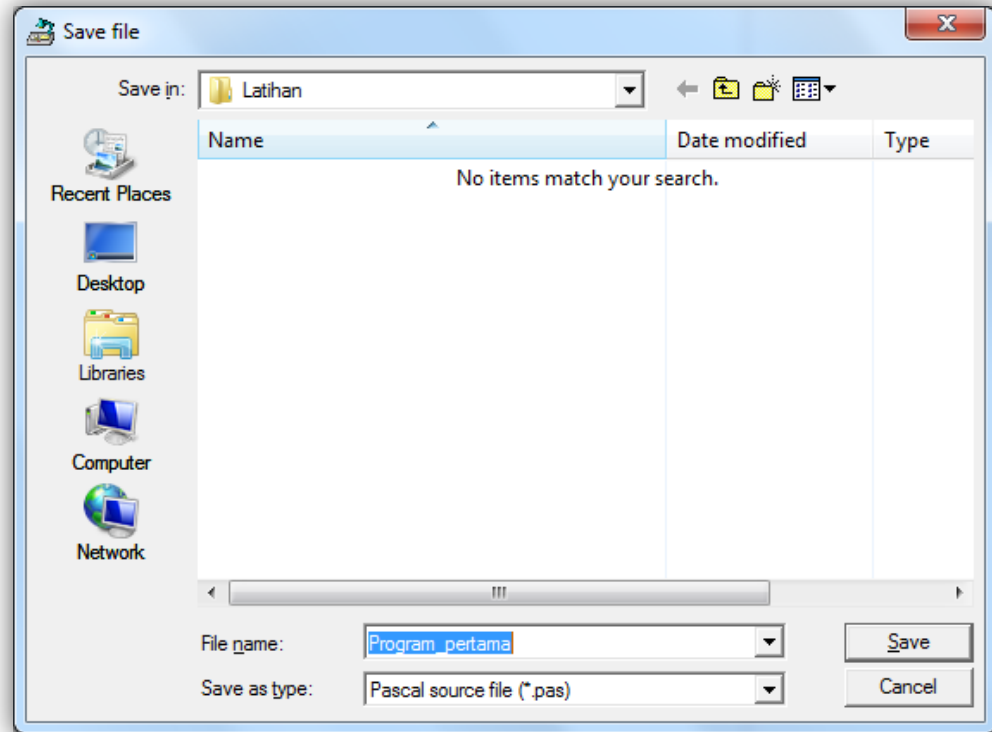
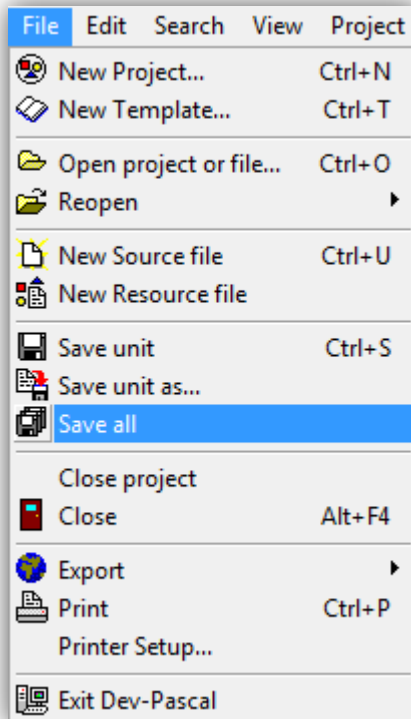
Save the project in the folder that had been provided

Step 6



If you have done with all steps **correctly**, you will get this
view on your computer

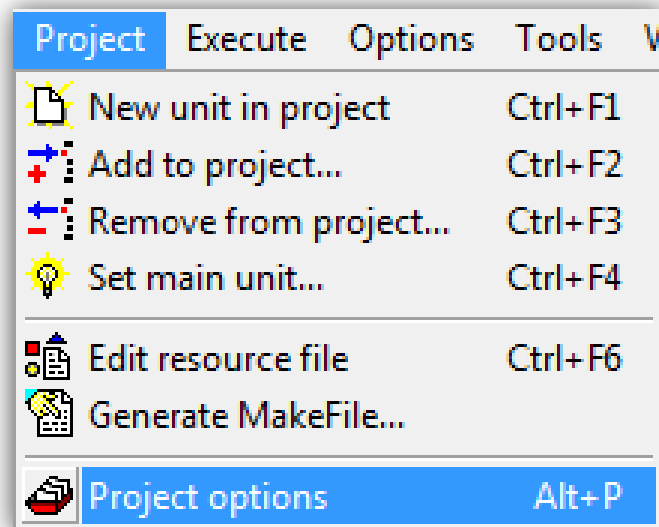
Step 7



Save this file in the **same folder** that contains the project

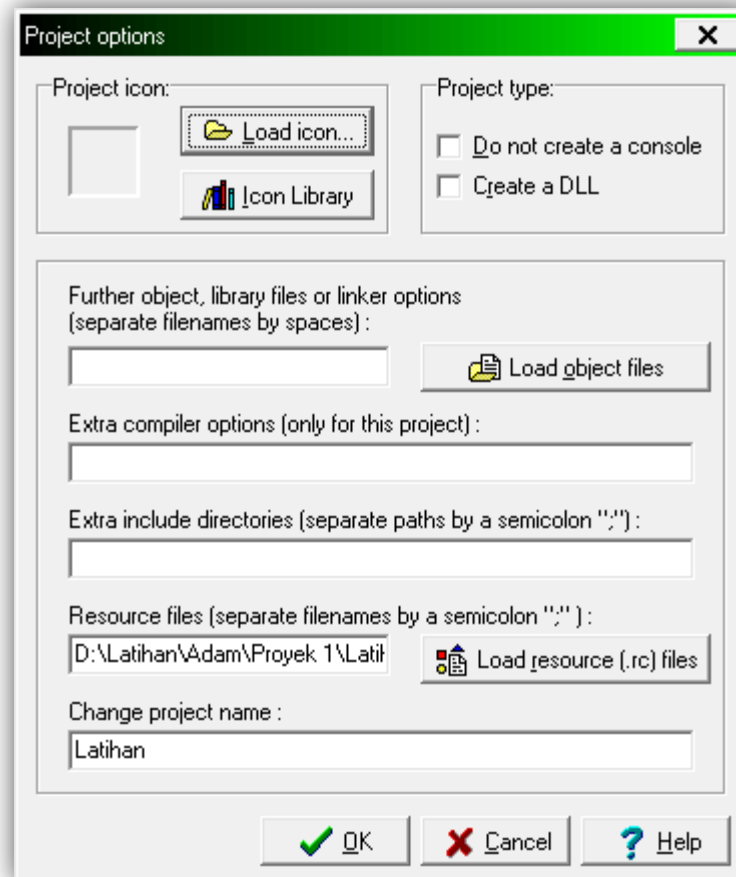
Step 8

Give an icon to your project. Click **Project → Project options** in menu bar.



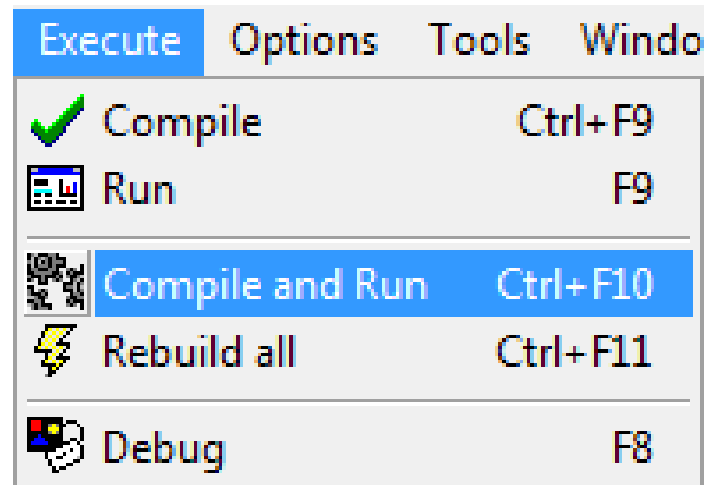
WARNING: Icon is an mandatory thing in Dev Pascal project

Step 9



Click **Load icon** then choose an icon that you want. Click **OK** to finish this step.

Step 10



Type **pascal syntax** then click **CTRL + F10** or click **Execute** → **Compile and Run** to see the result of this program.

Algorithm Notation VS Pascal Notation

VS



Example of Algorithm Notation

1	{ ini adalah notasi algoritma } → komentar
2	
3	Algoritma judul_algoritma
4	{I.S.: →diisi keadaan yang terjadi di awal algoritma}
5	{F.S.: →diisi keadaan yang terjadi di akhir algoritma}
6	
7	<u>Kamus/Deklarasi:</u>
8	{diisi pendefinisian konstanta}
9	
10	{diisi deklarasi variabel beserta tipe data}
11	
12	<u>Algoritma/Deskripsi:</u>
13	{diisi dengan input, proses, dan output}

Example of Pascal Notation

1	{ ini adalah notasi pascal} → komentar
2	program judul_program;
3	
4	var
5	{diisi pendefinisian konstanta}
6	
7	{diisi deklarasi variabel beserta tipe data}
8	
9	begin
10	{diisi dengan input, proses, dan output}
11	end.

Algorithm Notation VS Pascal Notation

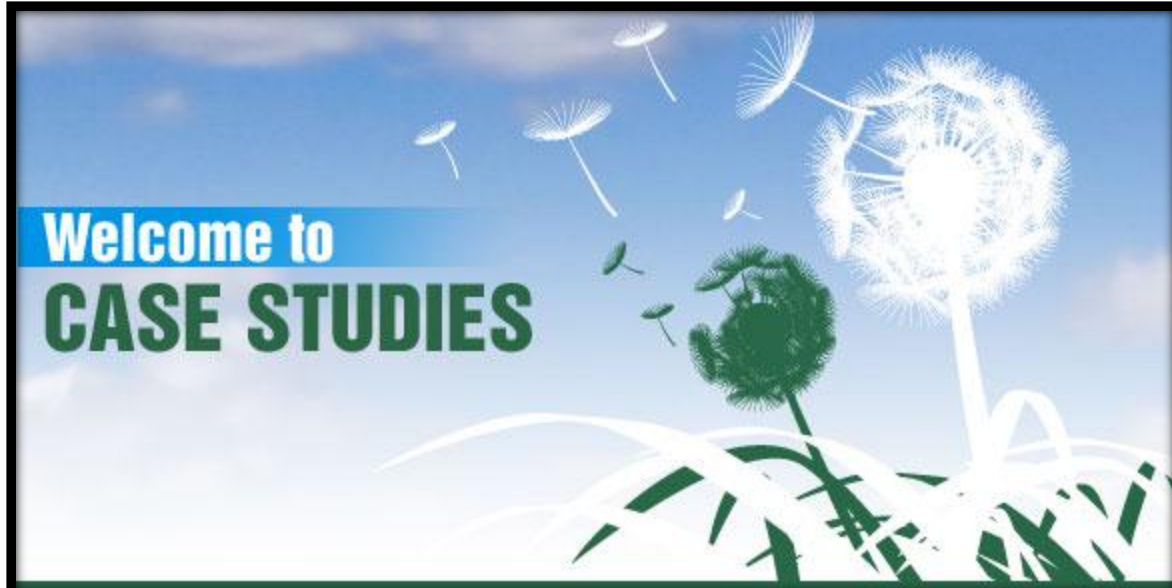
Num	ALGORITHM	PASCAL
1	<u>Kamus</u> :	var
2	<u>Algoritma</u> :	begin end.
3	<u>input</u> (variabel)	readln(variabel); read(variabel);
4	<u>output</u> ('.....')	write('.....'); atau writeln('.....');
5	<u>output</u> ('.....',variabel)	write('.....',variabel); atau writeln('.....',variabel);
6	<u>output</u> (variabel)	write(variabel); atau writeln(variabel);
7	←	:=

Your First Pascal Program

```
1  program Program_Pertama;
2  uses crt; {pemanggilan unit crt untuk readkey()}
3
4  begin
5      writeln('Selamat Datang');
6      write('Di');
7      writeln(' UNIKOM');
8      writeln('Bandung');
9      writeln();
10     write('Tekan sembarang tombol untuk menutup. ');
11     readkey();
12 end.
```




Welcome to
CASE STUDIES



Exchange value with additional variabel (Algorithm)

Algoritma Tukar_Nilai

{I.S.: Nilai variabel a dan b dimasukkan oleh user}

{F.S.: Menampilkan hasil penukaran nilai variabel a dan b}

Kamus:

a,b: integer

bantu: integer

Algoritma:

output('Masukkan nilai a: ')

input(a)

output('Masukkan nilai b: ')

input(b)

bantu \leftarrow a

a \leftarrow b

b \leftarrow bantu

output('Nilai a sekarang : ',a)

output('Nilai b sekarang : ',b)

Exchange value with additional variabel (Pascal)

```
1  program Tukar_Nilai;
2  uses crt; {pemanggilan unit crt untuk readkey()}
3
4  var
5      a,b:integer;
6      bantu:integer;
7
8  begin
9      write('Masukan nilai a: '); readln(a);
10     write('Masukan nilai b: '); readln(b);
11     bantu:=a;
12     a:=b;
13     b:=bantu;
14     writeln('Nilai a sekarang: ',a);
15     writeln('Nilai b sekarang: ',b);
16     readkey();
17 end.
```


Exchange value without additional variabel (Algorithm)

```
1  Algoritma Tukar_Nilai
2  {I.S.: Nilai variabel a dan b dimasukkan oleh user}
3  {F.S.: Menampilkan hasil penukaran nilai variabel a dan b}
4
5  Kamus:
6    a,b: integer
7
8  Algoritma:
9    input(a,b)
10   a←a+b
11   b←a-b
12   a←a-b
13   output('Nilai a sekarang : ',a)
14   output('Nilai b sekarang : ',b)
```


Exchange value with additional variabel (Pascal)

```
1  program Tukar_Nilai;
2  uses crt; {pemanggilan unit crt untuk readkey()}
3
4  var
5      a,b:integer;
6
7  begin
8      write('Masukan nilai a: '); readln(a);
9      write('Masukan nilai b: '); readln(b);
10     a:=a+b;
11     b:=a-b;
12     a:=a-b;
13     writeln('Nilai a sekarang: ',a);
14     writeln('Nilai b sekarang: ',b);
15     readkey();
16 end.
```



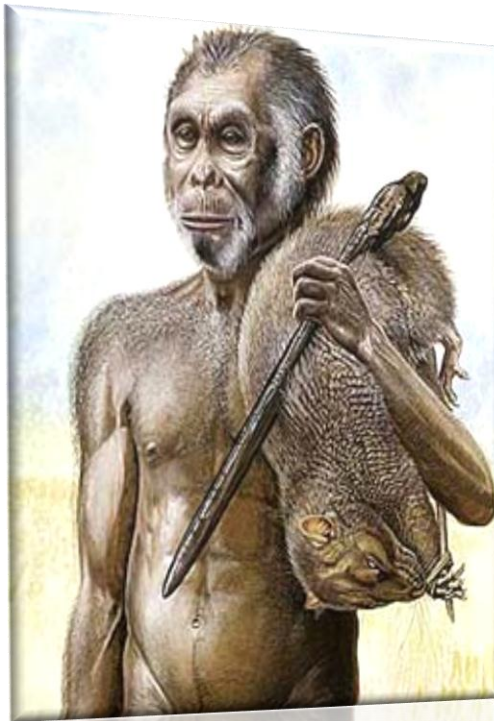

Data Type

Data Type in Algorithm and Pascal

- Tipe Data Dasar (Predefined Data Type)
- Tipe Data Bentukan (user-defined Data Type)



Predefined Data Type



- **Have been known** in daily life.
- **Such as:** logic number, integer, real number, characters, and string.

Operation in Logic Number

x	<u>not</u> x
true	false
false	true

x	y	x <u>and</u> y	x <u>or</u> y	x <u>xor</u> y
true	true	true	true	false
true	false	false	true	true
false	true	false	true	true
false	false	false	false	false

Integer

- Name : integer
- Value : - (\sim) until + (\sim) (without .)
- Arithmetic : +, -, *, /, div, mod
- Comparison : <, \leq , >, \geq , =, \neq .



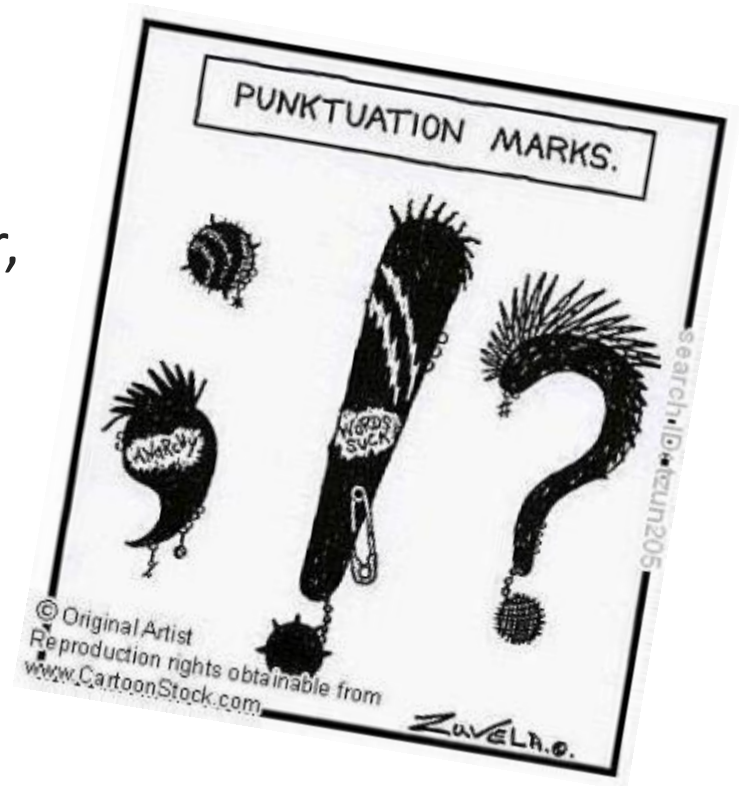
Real

- Name : real
- Value : - (\sim) until + (\sim)
- Arithmetic : +, -, *, /
- Comparison : <, \leq , >, \geq , =, \neq .



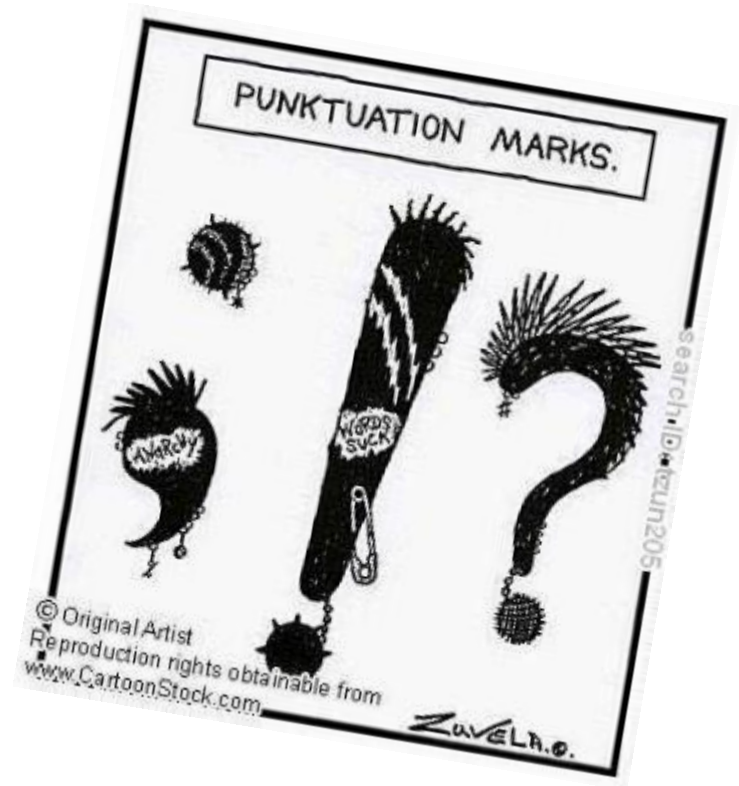
Characters

- Name : **char**
- Value : all alphabet, decimal number, punctuation mark, arithmetic operator, and ASCII
- Comparison : $<$, \leq , $>$, \geq , $=$, \neq .



String

- Name : **String**
- Value : set of characters (flanked with ' ')
- Comparison : $<$, \leq , $>$, \geq , $=$, \neq .



User-defined Data Type

- Predefined Data Type that was named with a new one.
- Structure type.



Modified Predefined Data Type

- **Reason** : Easy to remember and **High readability**.
- **Keyword** : **type**
- **Example**:

type

pecahan : **real** { : can be replaced with = }

Structure Type

- **Reason** : set of data that have different data type.
- **Example** :

type

Mahasiswa = record

< NIM : integer, {0..9}

Nama : string, {'A'..'Z', 'a'..'z'}

Nilai : real {0..100} >

Structure Type

- If **mhs1** is **mahasiswa type**, so to access each field in mhs1

can be done with these statement:

- a. mhs1.NIM
- b. mhs1>Nama
- c. mhs1.Nilai

Data Type in Algorithm and Pascal

Algorithm	Pascal	Range in Pascal
<u>boolean</u>	boolean	true dan false
<u>integer</u>	byte	0..255
	shortint	-128..127
	word	0..65535
	integer	-32768..32767
	longint	-2147483648..2147483647
<u>real</u>	real	$2.9 \times 10^{-39} \dots 1.7 \times 10^{38}$
	single	$1.5 \times 10^{-45} \dots 3.4 \times 10^{38}$
	double	$5.0 \times 10^{-324} \dots 1.7 \times 10^{308}$
	extended	$3.4 \times 10^{-4932} \dots 1.1 \times 10^{4932}$
<u>char</u>	char	
<u>string</u>	string	
	string[n]	
<u>type</u> varrecord: <u>record</u> < field1:type1, field2:type2, ... field_n:type_n >	type	
	varrecord=record field1:type1; field2:type2; ... field_n:type_n; end;	

Operator in Algorithm and Pascal

Algorithm	Pascal
+	+
-	-
*	*
/	/
<u>div</u>	div
<u>mod</u>	mod

Algorithm	Pascal
<	<
≤	<=
>	>
≥	>=
=	=
≠	<>

Algorithm	Pascal
<u>not</u>	not
<u>and</u>	and
<u>or</u>	or
<u>xor</u>	xor

Algorithm	Pascal
<u>type</u>	type
<u>const</u>	const
<u>true</u>	true
<u>false</u>	false
{ komentar }	{ komentar } (* komentar *)



Identifier

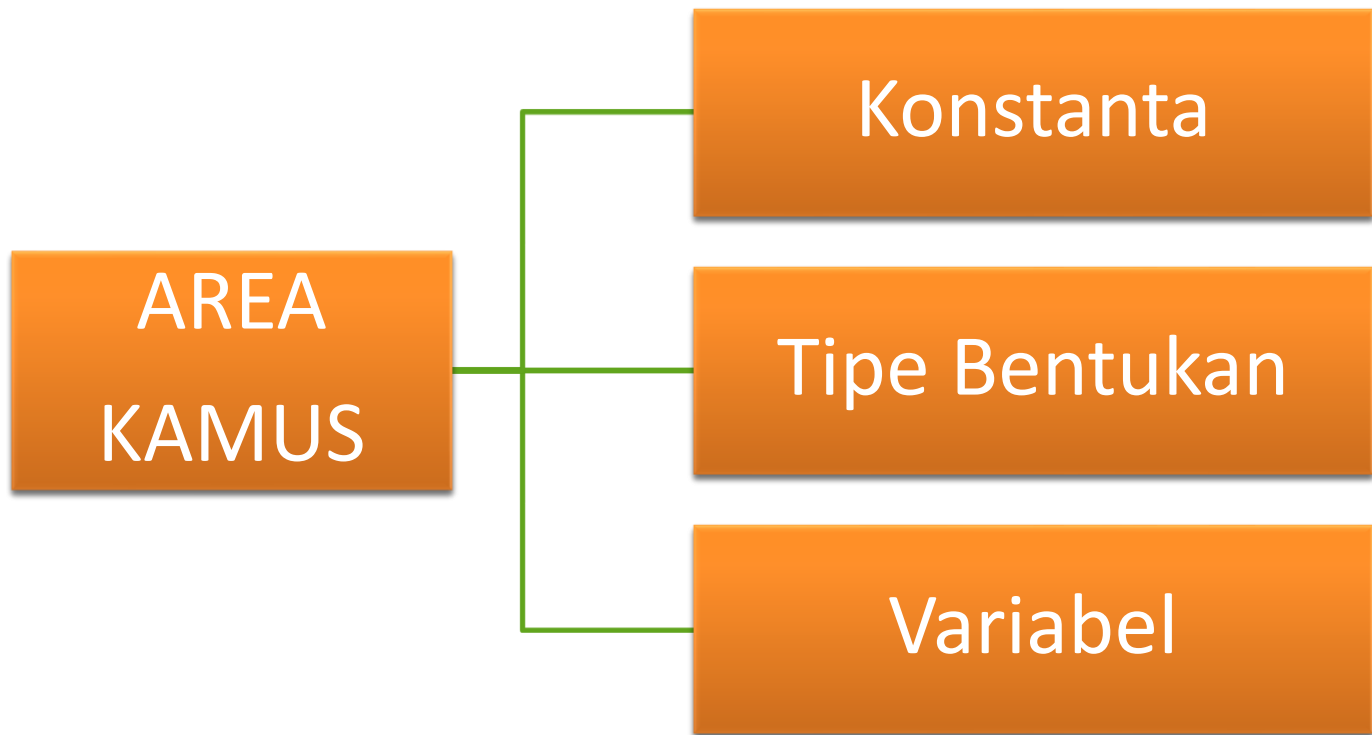
Definition, Rules and Expression

Definition of Identifier

Identifiers can be used to access something in algorithm or program.



Definition of Identifier



Rules of Naming

- Name must be started with alphabet.
- Upper case and lower case are the same thing in Pascal (case insensitive) → Suggest: should be consistent.
- Name only consists of alphabet, number, and underscore.
- Identifier can't contain arithmetic operator, relational, and punctuation mark, space.
- Choose the name that easy to remember.

Variable VS Constants

- Variable and Constants was used to store the value in memory.
- Variable can change the value in the middle of running time.
- Constants will keep the value permanently while running time.

Variable VS Constants

Variable Declaration

```
Nama_variabel:tipe_data
```

```
Example: x,y:integer
```

Constants Declaration

```
type
```

```
const nama_konstanta = nilai_konstanta
```

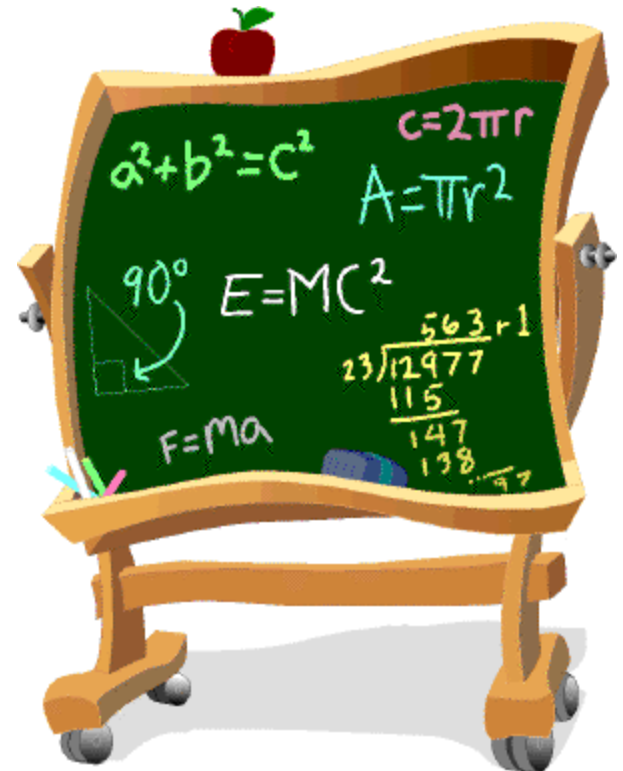
```
Contoh:
```

```
type
```

```
const phi =3.14
```


Math and Algorithm Notation

- Prefix $\rightarrow *79$, $*+a/bc-d*ef$
- Infix $\rightarrow 7*9$, $a+b/c*d-e*f$
- Postfix $\rightarrow 68*$, $abc/+def*-*$



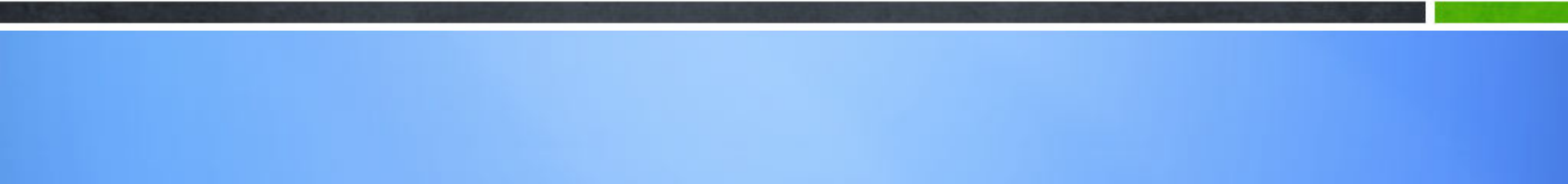
Math and Algorithm Notation

- $luas = \frac{1}{2}(alas.tinggi) \rightarrow luas \leftarrow 1/2 * (alas * tinggi)$
- $a = \frac{10b + 3c}{5d} \rightarrow a \leftarrow (10 * b + 3 * c) / (5 * d)$





EXERCISE



Exercise 1

Declare user-defined data type for cases:

- SIM
- KTP
- Lecturer Data

Exercise 2

Convert these math notations into algorithm notations:

- $m = \frac{a-b}{3ac} \left(1 - \frac{bcd}{fgh} \right)$
- $x = \frac{-b + 2c^2 + 4abc}{2c(3a + 4c)}$



THANK YOU

GRACIAS

Contact Person:

Adam Mukharil Bachtiar
Informatics Engineering UNIKOM
Jalan Dipati Ukur Nomor. 112-114 Bandung 40132
Email: adfbipotter@gmail.com
Blog: <http://adfbipotter.wordpress.com>