

**PROG0101**  
**FUNDAMENTALS OF PROGRAMMING**

**Chapter 4**  
**Variables and Data Types**

## Topics

- Variables
- Constants
- Data types
- Declaration

## Variables

- A symbol or name that stands for a value.
- A **variable** is a value that can change.
- Variables provide **temporary storage** for information that will be needed during the lifespan of the computer program (or application).

## Variables

Example:

$$z = x + y$$

- This is an example of programming expression.
- x, y and z are variables.
- Variables can represent numeric values, characters, character strings, or memory addresses.

## Variables

- Variables store everything in your program.
- The purpose of any useful program is to modify variables.
- In a program every, variable has:
  - Name (Identifier)
  - Data Type
  - Size
  - Value

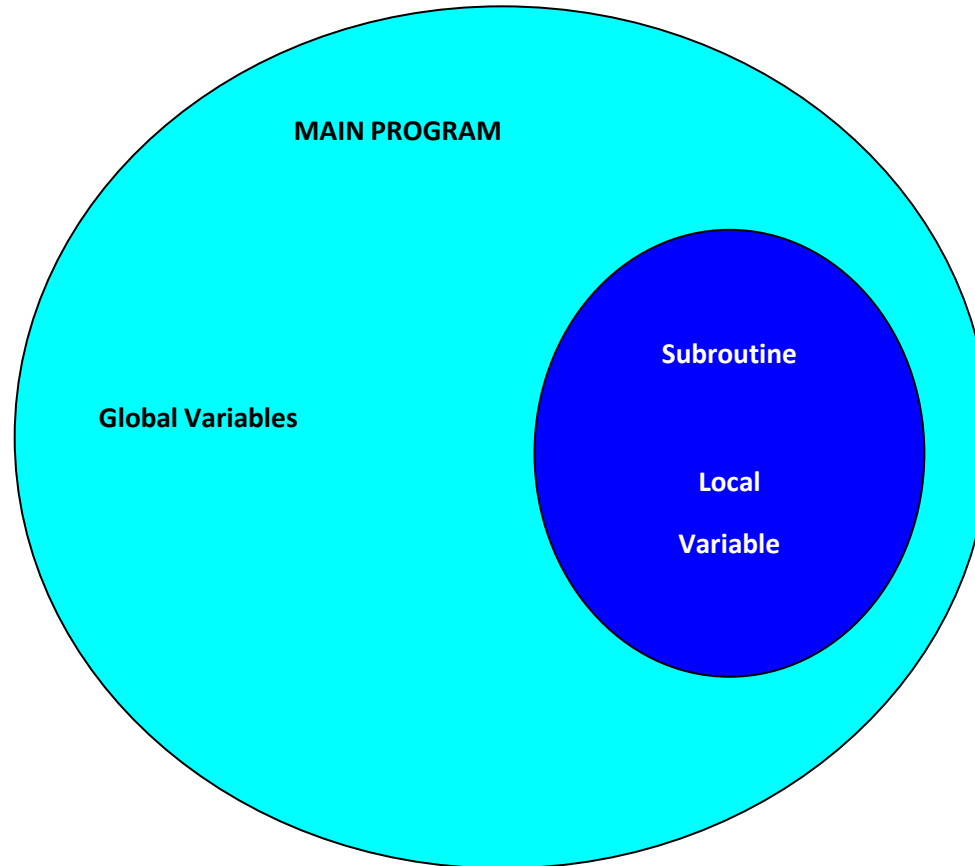
## **Types of Variable**

- There are two types of variables:
  - Local variable
  - Global variable

## Types of Variable

- **Local variables** are those that are in scope within a specific part of the program (function, procedure, method, or subroutine, depending on the programming language employed).
- **Global variables** are those that are in scope for the duration of the programs execution. They can be accessed by any part of the program, and are read-write for all statements that access them.

## Types of Variable





## Rules in Naming a Variable

- There are certain rules in naming variables (identifier).
- They are:
  - It should use only alphabets, numbers and underscore ( \_ )
  - It should not begin with a number and must have at least one alphabet.
  - It should not be a reserved word.
- Examples of reserved words:

main	long	if	do	continue
short	else	return	const	int
double	break	void	while	char

## Rules in Naming a Variable

- Following are some valid identifiers:

x          number1          a123          number\_2          \_xyz

- Following are some invalid identifiers:

print main	space not allowed
123number	name should not begin with number
main	reserved word

## Exercise

Explain these mathematical problems by using variables.

1. Area of square
2. Area of triangle
3. Area of circle
4. Average speed

## Constants

- The opposite of a variable is a constant.
- A constant is a value that **never changes**.
- Because of their inflexibility, constants are used less often than variables in programming.
- A constant can be :
  - a number, like 25 or 3.6
  - a character, like *a* or \$
  - a character string, like "this is a string"

## Constants

- Example:
  - Calculate area of circle

$$\text{area} = 3.142 * r * r$$

- Variable:
  - area, r
- Constant
  - 3.142

## **Data Types**

- Data type is classification of a particular type of information.
- Data types are essential to any computer programming language.
- Without them, it becomes very difficult to maintain information within a computer program.
- Different data types have different sizes in memory depending on the machine and compilers.

## **Data Types**

- Integer
- Floating-point
- Character
- String
- Boolean

## Integer

- A whole number, a number that has no fractional part.
- The following are integers:

0      1      -125      144457

- In contrast, the following are not integers:

5.34    -1.0    1.3E4    "string"

- There are often different sizes of integers available; for example, PCs support short integers, which are 2 bytes, and long integers, which are 4 bytes.



## Floating-point

- A number with a decimal point.
- The following are floating-point numbers:

3.0    -111.5

- Floating-point representations are slower and less accurate than fixed-point representations, but they can handle a larger range of numbers.
- It require more space.

## Character

- In graphics-based applications, the term **character** is generally reserved for letters, numbers, and punctuation..
- It only can represent single character.
- The following are characters:

a      A      @      \$      2

## String

- String is a sequence of character.
- Example:
  - Kuala Lumpur
  - John White
  - Computer

## Boolean

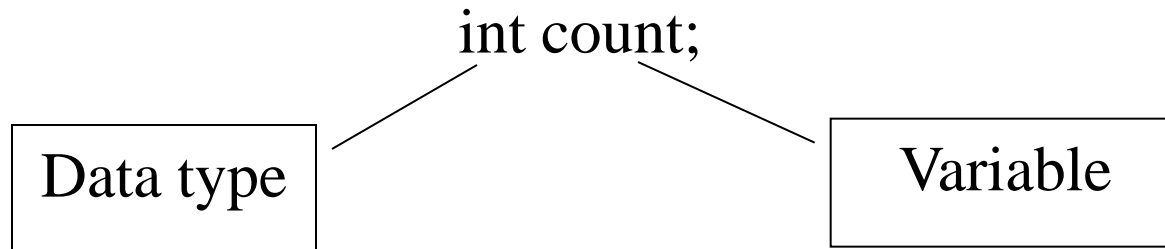
- This data type only have two possible values:
  - True
  - False
- Use this data type for simple flags that track true/false conditions.

## Declaration

- In programming languages all the variables that a program is going to use must be **declared** prior to use.
- Declaration of a variable serves two purposes:
  - It associates a **type** and an identifier (or name) with the variable. The type allows the compiler to interpret statements correctly.
  - It allows the compiler to decide how much storage space to allocate for storage of the value associated with the identifier and to assign an address for each variable which can be used in code generation.

## Declaration

- A typical set of variable declarations that might appear at the beginning of a program could be as follows:



## Declaration

- Often in programming numerical constants are used, e.g. the value of  $\pi$  (pi = 3.14).
- It is well worthwhile to associate meaningful names with constants.
- These names can be associated with the appropriate numerical value in a **constant declaration**.
- The names given to constants must conform to the rules for the formation of identifiers as defined above.

## Declaration

- The following constant declaration:

```
const int days_in_year = 365;
```

- The general form of a constant declaration is:

```
const type constant-identifier = value ;
```



## Exercise

Are these identifiers valid or not? Explain.

1. a1
2. Number3
3. 123abc
4. a\_123
5. no 2

## Exercise

Identify the data type:

1. 312
2. 2.153
3. a1
4. a
5. True
6. hello