CSE101-Lec 3

Constant

Variable

Expression

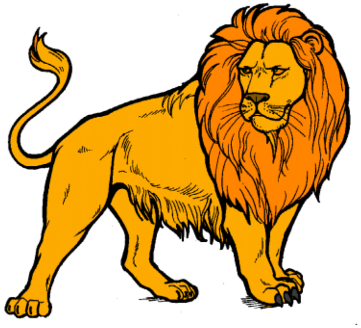
Outline

- In this lecture we will study
  - Constant
  - Variable
  - Expression

Tokens

- We have seen that Tokens are broadly classified as:
  - Identifiers
  - Keywords
  - **Constants**
  - **Variables**
  - Strings
  - Operators
  - Special character

Constants

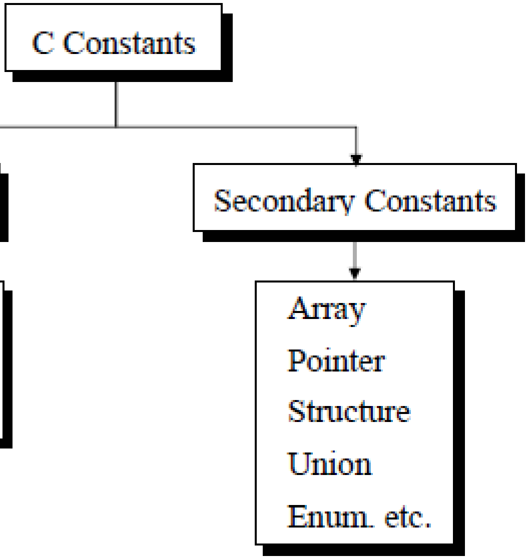


= Lion ≠

= Spanner ≠

Constants

- The entity which do not change throughout the execution are called constants.



- Types of Primary constants:
  - Integer constant
  - Character constant
  - Floating point constants
  - String constants

Name of person remains same through out the life, example: Amit, Shubham, etc.

- **Character constants**
  - Constants enclosed in single quotes(‘ ’).
  - It can be any letter from character set.

Example : ‘\n’, ‘\t’ or ‘a’

- **String Constants**
  - Set of zero or more characters enclosed in double quotes (eg: “ ”)
  - It is represented as sequence of characters within double quotes.

Example : “This is C programming”

- **Integer Constants**
  - When the constant contains only digits without any decimal part

Example : 5;
-987;

- **Floating Constant**
  - Constants that contains number with decimal points

Example : 3.14;
309.89

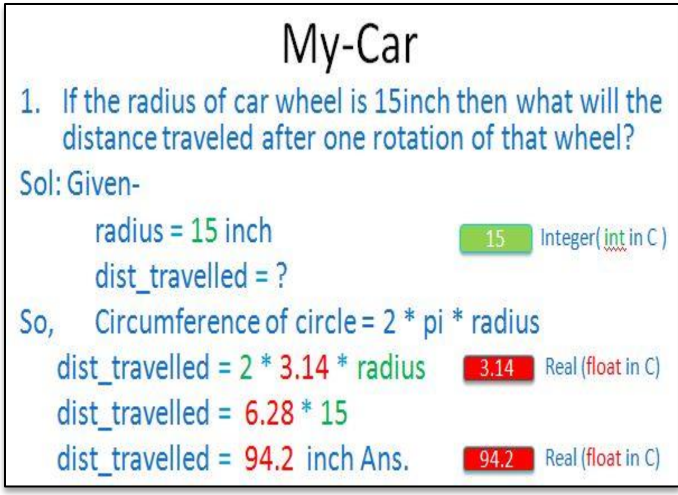
My-Car

In My-Car problem the constant value is 3.14 which is the value of pi and always same.

- pi = 3.14

Therefore:

dist_travelled = 2 * pi * radius.



My-Car

1. If the radius of car wheel is 15inch then what will the distance traveled after one rotation of that wheel?

Sol: Given-

radius = 15 inch — 15 Integer(int in C)

dist_travelled = ?

So, Circumference of circle = 2 * pi * radius

dist_travelled = 2 * 3.14 * radius — 3.14 Real (float in C)

dist_travelled = 6.28 * 15

dist_travelled = 94.2 inch Ans. — 94.2 Real (float in C)

➢ pi is a floating point constant.

Variables

Animal =

Tool =

Variables

- Variable is an entity which may change.
- Variable is used to hold result and reserve memory for the data.

Syntax

```
datatype variable_name;
```

The naming of variable is done by following the same rules of identifier naming.

Eg. What is your **hobby**?
The answer could be **reading, dancing, drawing**, etc.
So the answer to such questions may change during the life time of the person

Rules for naming a Variable

1. An variable name is any combination of 1 to 31 digits: alphabets , digits or underscores.

2. The first character in the variable name must be an alphabet or underscore.

3. No blanks or special symbol other than an underscore can be used in an variable name.

4. Keywords are not allowed to be used as variables.

Variables

Let us build some variables:

For speed of car we need to know

- Distance traveled
- Time taken to travel the distance

Variables to be declared as

- Speed, **s1**, speed_of_car
- Distance, **d1**, dist
- Time, **t1**, time_of_travel

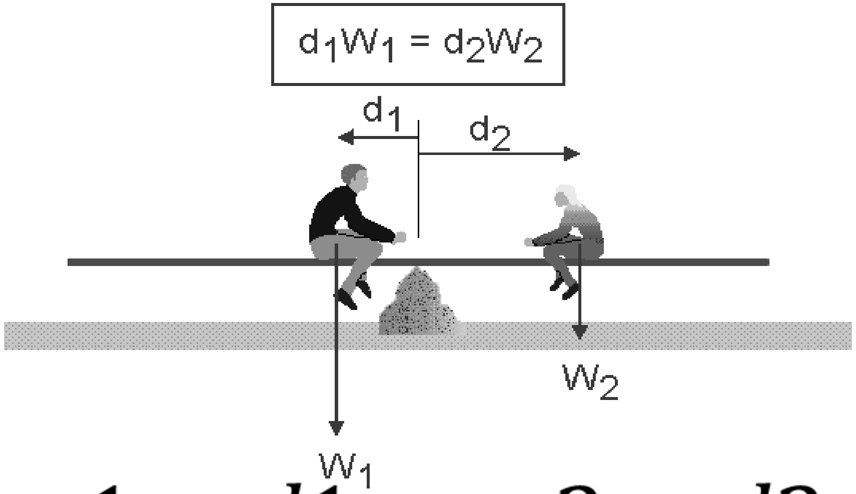
$$s1 = d1 \div t1$$

See-Saw

- A bit more complex situation see-saw

Variables to be declared as

- Weight_boy, **w1**
- Distance_boy, **d1**
- Weight_girl, **w2**
- Distance_girl, **d2**



$$w1 \times d1 = w2 \times d2$$

$$wb \times db = wg \times dg$$

- It is to be assessed that at what distance 50Kg girl should sit in order to balance a boy of 70Kg sitting 2m away from the center ‘o’

Variable Initialization

- Assigning some value to the variable at time of creation of variable is known as variable initialization.

Syntax

```
datatype variable_name =  value;
```

```
Example:  int radius= 15;
          float pi = 3.14;
          char grade = ‘A’;
```

Expressions

- Expressions are the statements or the instruction given to computer to perform some operation.
- Every expression results in some value that can be stored in a variable.
- Following are few example of expressions in program:
  - Expression to calculate speed of a car.
    - Speed=distance/time
  - Expression to find similarity of two things.
    - c=value1>value2

- Expressions in C are basically **operators** acting on **operands**.
- An **operand** is an entity on which operation is to be performed.

Example: addition of two numbers, 5+8, these numbers will be operands.

- An **operator** specifies the operation to be applied on operands.

Example: The addition, subtraction, etc will be operators

- Expressions are made of one or more operands.
- Statements like :

  a = b + c,

  ++z

  300 > (8 * k)

Types of Expressions

- The type of expression depend upon the type of operator used in the expression.
- It can be:
  - Arithmetic operators.

    3 + 6 = 9

    4 * 2 = 8
  - Relational or logical operators.

    height_boy>=height_girl
  - Increment and decrement operator.

    count=count++

After this : Operators
Types of operators

cse101@lpu.co.in