Freebooks.pk

Free Notes • Past Papers • Guides for Students

2nd Year Computer Science — Punjab Board

CHAPTER 9

Elements of C

Key Concepts · Definitions · Short & Long Questions · MCQs · Diagrams · Exam Tips

This chapter covers the Elements of C from the 2nd Year (ICS Part-II) Computer Science syllabus of the Punjab Curriculum and Textbook Board (PTB/PCTB). It explains variables and constants, data types, operators, expressions and type conversion, the basic building blocks of every C program. These notes are prepared by freebooks.pk.

Before writing real programs, we must know the elements from which they are built. This chapter explains variables, constants, data types and operators in C.

Learning Objectives

- Define a variable and a constant.
- State the basic data types of C.
- Declare and initialise variables.
- Describe the arithmetic, relational, logical and assignment operators.
- Form and evaluate expressions.
- Explain type conversion.

Key Concepts

Variables

A variable is a named location in the computer's memory used to store a value that can change while the program runs. Every variable has a name (an identifier), a data type that decides what kind of value it can hold, and a value. Before it is used, a variable must be declared, for example int age; which reserves memory for an integer called age. A variable can also be initialised, that is, given a starting value, as in int age = 20;.

Constants

A constant is a value that does not change while the program runs. Constants may be numbers (such as 100 or 3.14), single characters written in single quotes (such as 'A'), or strings of characters written in double quotes (such as "Pakistan"). A named constant can be created using #define or the const keyword, for example #define PI 3.14, so that the fixed value can be used by name throughout the program.

Data Types

The data type of a variable tells C what kind of value it will store and how much memory to use. The basic data types in C are int, for whole numbers such as 5 or -20; float, for numbers with a decimal part such as 3.14; double, for decimal numbers needing greater precision; and char, for a single character such as 'A'. Choosing the correct data type ensures the data is stored correctly and efficiently.

Arithmetic Operators

Operators are symbols that perform operations on values. The arithmetic operators carry out calculations: + (addition), - (subtraction), * (multiplication), / (division) and % (modulus,

which gives the remainder of a division). For example, 7 % 3 gives 1. These operators, combined with variables and constants, are used to form arithmetic expressions such as a + b * c.

Relational and Logical Operators

Relational operators compare two values and give a result that is true or false; they are < (less than), > (greater than), <= (less than or equal), >= (greater than or equal), == (equal to) and != (not equal to). Logical operators combine such conditions: && (AND, true only if both are true), || (OR, true if either is true) and ! (NOT, which reverses a condition). These operators are used to make decisions in a program.

Assignment Operators and Expressions

The assignment operator = stores the value on its right into the variable on its left, as in sum = a + b;. C also provides shorthand assignment operators such as += , -= , *= and /=; for example, x += 5; means x = x + 5;. An expression is a combination of variables, constants and operators that produces a value, such as (a + b) / 2; the computer evaluates the expression according to the rules of operator precedence.

Type Conversion

Type conversion is the changing of a value from one data type to another. It may happen automatically (implicit conversion), for example when an int is used where a float is expected, or it may be done deliberately by the programmer (explicit conversion or type casting), by writing the required type in brackets, as in (float)a. Type conversion is important to make sure that calculations, especially division, give the correct kind of result.

Important Definitions

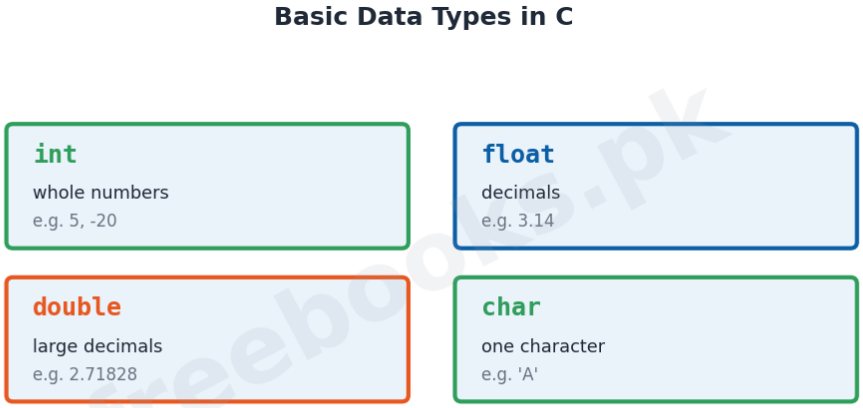
Term	Definition
Variable	A named memory location whose value can change.
Constant	A value that does not change during the program.
Data type	The kind of value a variable can store (int, float, char, double).
Declaration	Stating a variable's name and type before use (e.g. int age;).
Operator	A symbol that performs an operation on values.
Expression	A combination of variables, constants and operators giving a value.
Modulus (%)	An operator giving the remainder of a division.
Type conversion	Changing a value from one data type to another.

Key Facts & Rules

Item	Detail
Declare a variable	int age; (type name;)
Initialise	int age = 20;
Data types	int, float, double, char
Arithmetic	+ - * / % (remainder)
Relational	< > <= >= == !=
Logical	&& || !
Shorthand assignment	x += 5; means x = x + 5;

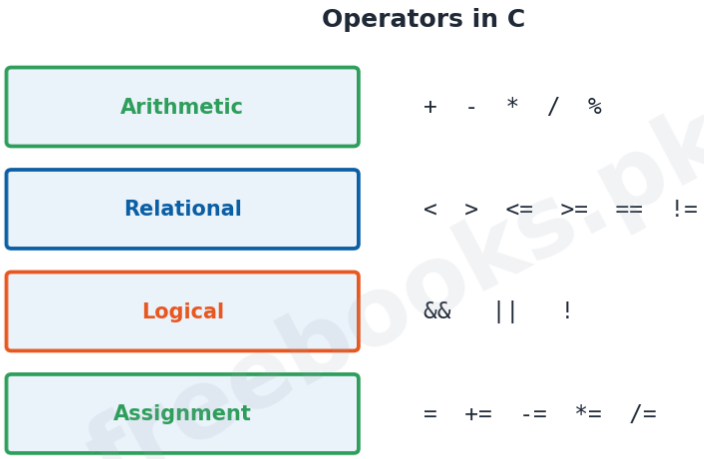
Diagrams & Illustrations

Basic data types in C: the basic C data types: int (whole numbers), float and double (decimals) and char (a single character).



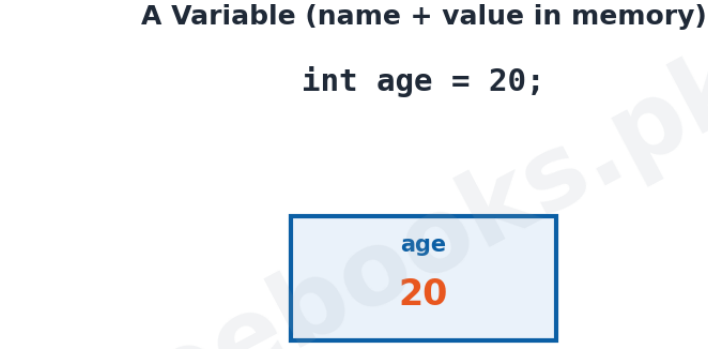
Basic data types in C

Operators in C: the groups of operators in C: arithmetic, relational, logical and assignment.



Operators in C

A variable: a variable shown as a named memory location holding a value (int age = 20).



A variable

Solved Examples & Practice

Declare and initialise

int marks = 85; declares an integer variable called marks and gives it the value 85.

Modulus

The expression 10 % 3 gives 1, the remainder when 10 is divided by 3.

Relational result

The expression 5 > 3 is true, and 5 == 3 is false.

Shorthand

If x is 10, then x += 5; makes x become 15.

Short Questions & Answers

Q1. What is a variable?
Ans. A named location in memory used to store a value that can change while the program runs.

Q2. Differentiate a variable and a constant.
Ans. A variable's value can change during the program; a constant's value stays the same.

Q3. Name the basic data types of C.
Ans. int (whole numbers), float and double (decimals), and char (a single character).

Q4. What does the modulus operator (%) do?
Ans. It gives the remainder of a division, for example 7 % 3 gives 1.

Q5. Name the logical operators.
Ans. && (AND), || (OR) and ! (NOT).

Q6. What is type conversion?
Ans. Changing a value from one data type to another, either automatically or by type casting.

Long Questions & Answers

Q1: Explain variables, constants and data types in C with examples.

The most basic elements used to store information in a C program are variables and constants. A variable is a named location in the computer's memory that is used to hold a value which may change while the program is running. Every variable has three things: a name, which is an identifier chosen by the programmer, such as age or marks; a data type, which decides what kind of value the variable can hold; and a value, the actual data stored in it. Before a variable can be used it must be declared, which reserves memory for it and states its type, as in the declaration int age; and it may also be initialised, that is, given a starting value at the same time, as in int age = 20;. A constant, in contrast, is a value that does not change while the program runs. Constants may be integer or real numbers such as 100 or 3.14, single characters written between single quotes such as 'A', or strings of characters written between double quotes such as "Pakistan"; a named constant can be created with #define or const, for example #define PI 3.14, so that a fixed value can be referred to by a meaningful name. Closely tied to variables is the idea of the data type, which tells C what kind of value a variable will store and how much memory to give it. The basic data types are int, used for whole numbers such as 5 or -20; float, used for numbers that have a decimal part, such as 3.14; double, used for decimal numbers that need greater precision; and char, used for a single character such as 'A'. Choosing the correct data type for each variable ensures that data is stored accurately and that memory is used efficiently.

Q2: Describe the arithmetic, relational, logical and assignment operators in C.

Operators are the special symbols that tell C to perform operations on data, and C provides several groups of them. The arithmetic operators are used for calculations: the plus sign adds, the minus sign subtracts, the asterisk multiplies, the slash divides, and the percent sign, called the modulus operator, gives the remainder left after a division, so that 10 % 3 gives 1. Using these, the programmer can build arithmetic expressions such as a + b * c. The relational operators are used to compare two values, and each gives a result that is either true or false: less-than, greater-than, less-than-or-equal, greater-than-or-equal, the double equals sign for is-equal-to, and the exclamation-equals for is-not-equal-to. For example, the comparison 5 > 3 is true while 5 == 3 is false. The logical operators are used to combine such conditions: the double ampersand means AND and is true only when both conditions are true; the double vertical bar means OR and is true when at least one condition is true; and the exclamation mark means NOT and reverses the truth of a condition. Relational and logical operators together allow a program to test conditions and make decisions. Finally, the assignment operators are used to store values in variables. The single equals sign is the basic assignment operator, which copies the value on its right into the variable on its left, as in sum = a + b;. C also offers shorthand assignment operators such as +=, -=, *= and /=; for instance x += 5; is a short way of writing x = x + 5;. Together these four groups of operators provide everything needed to calculate, compare, combine and store data in a program.

Q3: Explain expressions and type conversion in C.

An expression in C is any valid combination of variables, constants and operators that the computer can evaluate to produce a single value; for example a + b, (a + b) / 2 and marks > 50 are all expressions. When the computer evaluates an expression that contains several operators, it does not simply work from left to right but follows the rules of operator precedence, which decide the order in which the operators are applied; for instance, multiplication and division are done before addition and subtraction, so that in the expression 2 + 3 * 4 the multiplication is performed first, giving 14, and brackets can be used to change this order when needed. Expressions are the means by which all calculations and tests in a program are carried out. Closely related to expressions is the idea of type conversion, which is the changing of a value from one data type to another. Type conversion can happen in two ways. Implicit conversion is done automatically by C when values of different types are mixed in an expression; for example, if an integer is used in a calculation that also involves a float, C automatically converts the integer to a float so that the arithmetic can be carried out correctly. Explicit conversion, also called type casting, is done deliberately by the programmer by writing the desired type in brackets in front of the value, as in (float)a, which treats the value of a as a float. Type conversion is important because it affects the result of calculations; a common example is integer division, where dividing one integer by another discards the fractional part, so a programmer who wants a decimal answer must convert one of the values to a float first. Understanding expressions and type conversion therefore helps a programmer to write calculations that give correct results.

MCQs with Answers

1. A named memory location whose value can change is a:
(a) constant (b) variable (c) keyword (d) operator
Correct Answer: (b) variable. A variable.

2. A value that does not change is a:
(a) variable (b) constant (c) function (d) loop
Correct Answer: (b) constant. A constant.

3. The data type for whole numbers is:
(a) float (b) int (c) char (d) double
Correct Answer: (b) int. int.

4. The data type for a single character is:
(a) int (b) char (c) float (d) string
Correct Answer: (b) char. char.

5. The operator that gives a remainder is:
(a) / (b) % (c) * (d) +
Correct Answer: (b) %. % (modulus).

6. Which is a relational operator?
(a) + (b) && (c) == (d) =
Correct Answer: (c) ==. == (equal to).

7. The logical AND operator is:
(a) && (b) || (c) ! (d) &
Correct Answer: (a) &&. &&.

8. x += 5; is the same as:
(a) x = 5 (b) x = x + 5 (c) x = x - 5 (d) x == 5
Correct Answer: (b) x = x + 5. x = x + 5;.

9. A combination of variables and operators giving a value is an:
(a) expression (b) identifier (c) array (d) header
Correct Answer: (a) expression. An expression.

10. Changing a value from int to float is:
(a) type conversion (b) assignment (c) declaration (d) comment
Correct Answer: (a) type conversion. Type conversion.

Quick Revision Summary

- Variable = named memory holding a changeable value; declare (int age;) then use.
- Constant = fixed value (numbers, 'A', "text"); #define PI 3.14.
- Data types: int, float, double, char.
- Operators: arithmetic (+ - * / %), relational (< > == !=), logical (&& || !), assignment (= += ...).
- Expression = variables + constants + operators -> a value (operator precedence).

- Type conversion: implicit (automatic) or explicit ((float)a) casting. Notes by freebooks.pk.

Exam Tips

- Define variable and constant with examples.
- List the four basic data types and their uses.
- Show a declaration and an initialisation.
- Give each operator group with its symbols.
- Explain the modulus operator with an example.
- Describe implicit and explicit type conversion.