Unit no. 01: Introduction to Programming

Exercise:

Define the following:

1. IDE Integrated Development Environment:

A software that provides a programming environment to facilitate programmers in writing and executing computer programs is known as an integrated Development environment (IDE). An IDE has Graphical User Interface GUI. An IDE consists of tools that help a programmer throughout the phases of writing, executing and testing in a computer program.

2. Compiler:

A compiler is a software that is responsible for conversion of a computer program written in some high-level language to machine language code.

3. Reserved words:

Every programming language has a list of words that are predefined. Each word has its specific meaning already known to the compile. These words are known as reserved words or keywords. e.g. int, float, if, else etc.

4. Main section of a program:

Main section of a program consists of main () function. Every C program must contain a main () function, and it is the starting point of execution.

5. Char data type:

In C language, the char (short for character) data type is one of the basic data types used to store a single character literal. The char data type is used to store a single character. Internally, a char is stored as an integer value (ASCII value). It takes up just 1 byte of memory for storage.

Give short answers to the questions:

1. Why do we need a programming environment?

A **programming environment** is a **set of tools and software** that helps programmers write, test, debug, and run their code efficiently. It provides a **text editor or IDE (like Visual Studio Code, Code::Blocks, Dev C++, etc.)** to write code easily. A programming environment includes a **compiler or interpreter** to **convert human-readable code into machine code**. It allows you to **build (compile)** and **run (execute)** programs with a single click or command. Help identify and fix errors.

2. Write the steps to create a C program file in the IDE of your lab computer?

Using Dev C++ :

1. Open **Dev C++**.
2. Go to File → New → Source File.
3. Write your C program.
4. Save it with a .c extension (e.g., hello.c).
5. Press F9 to **compile and run**.

3. Describe the purpose of a compiler?

Purpose of a Compiler in C Programming:

A **compiler** is a special program that **translates source code written in a high-level programming language (like C)** into **machine code (binary/executable)** that the computer's processor can understand and execute. The compiler acts as a **bridge between the programmer and the machine**, transforming high-level C code into an optimized, executable program while ensuring correctness and performance.

4. List down five reserved words in C programming language.

Five Reserved Words in C Programming Language:

Reserved words (also called **keywords**) are predefined words in C that **have special meaning** and **cannot be used as identifiers (like variable names)**.

Here are 5 commonly used reserved words in C:

int	Declares an integer type variable or function return type.
return	Exits from a function and optionally returns a value.
if	Starts a conditional (decision-making) block.
while	Starts a loop that runs as long as a condition is true.
void	Specifies that a function returns no value.

5. Discuss the main parts of the Structure of a C program?

The structure of a C program can be divided into 3 main parts:

1. Link Section Or Header Section:

   While writing programs in C, we make extensive use of functions that are already defined in the language. But before using the existing functions, we need to include the files where these functions have been defined. These files are called header files.

   General structure:

   # include < header _ file name>

2. Main Section:

It consists of a main () function. Every C program must contain a main () function, and it is the starting point of execution.

3. Body of main () function:

The body of the main () is enclosed in the curly braces {}. All the statements inside these curly braces make the body of main function.

6. Why do we use comments in programming?

Comments are lines in a program that are **not executed** by the compiler. They are used **to make code easier to understand for humans** — whether for the original programmer, teammates, or future developers. It improves Code Readability.

Types of Comments in C:

Type	**Syntax**	**Example**
Single-line	// comment	// This is a comment
Multi-line	/* comment block */	/* This explains the logic */

7. Differentiate between constants and variables?

Difference Between Constants and Variables in C:

Both **constants** and **variables** are used to store data in a C program, but they have **different behaviors**.

Here's a clear comparison:

Aspect	**Constants**	**Variables**
Definition	Fixed values that **do not change** during execution	Named storage locations whose **values can change**
Value Modification	Not allowed after initialization	Can be changed anytime in the program
Declaration Example	const int x = 10;	int x = 10;
Use Case	To store fixed values like π, tax rate, limits, etc.	To store values that change during the program
Keyword Used	const	No special keyword

8. Write down the rules for naming a variable.

Rules for naming a variable:

1. A variable name can only contain alphabets(uppercase or lowercase), digits and an underscore _ sign.
2. A variable name must begin with a letter or an underscore, it cannot begin with a digit.
3. A reserved word cannot be used as a variable name.
4. There is no strict rule on how long a variable name should be, but we should choose a concise length for variable name to follow good design practice.

9. Differentiate between char and int.

In C programming, both char and int are **data types**, but they are used for **different kinds of data**.

Aspect	char	int
Purpose	Used to store **single characters** (like 'A', 'z')	Used to store **whole numbers** (like 5, -20, 100)
Size (in memory)	Usually, **1 byte**	Usually **4 bytes** (can vary by system)
Data Stored	Character (actually stored as ASCII code internally)	Integer number
Example Declaration	char grade = 'A';	int marks = 85;
Format Specifier	%c in printf/scanf	%d in printf/scanf

10. How can we declare and initialize a variable?

Declaring and Initializing a Variable in C:

In C, **declaring** a variable means telling the compiler about the **name and data type** of the variable. **Initializing** a variable means **assigning it an initial value**. **Declaration** reserves memory. **Initialization** assigns value. We can do both together, or separately.

Syntax:

```
data_type variable_name;          // Declaration

data_type variable_name = value;   // Declaration + Initialization
```

Examples:

Data Type	Declaration	Initialization
Int	int age;	int age = 18;
Float	float price;	float price = 99.99;
Char	char grade;	char grade = 'A';