LECTURE NO. 1

PREVIEW OF TODAY'S TOPICS

☞ Program & Programming Languages

PROGRAM & PROGRAMMING LANGUAGES

LONG QUESTION

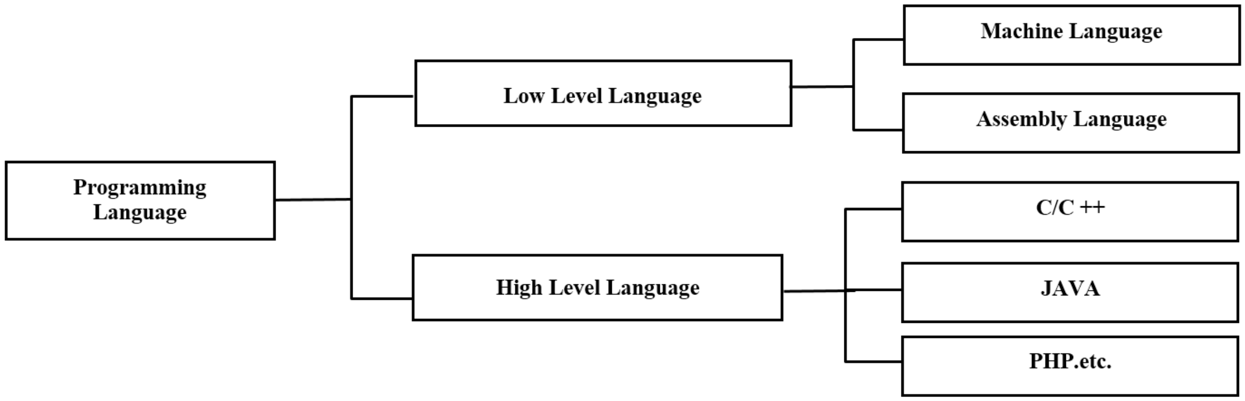
1. **What is programming language? Explain its types.** **(LHR 2022) (DGK 2022) (SWL 2021) (MTN 2021)**
2. **How would you define High Level Language? Write common characteristics of all high level program languages.** **(BWP 2022)(RWP 2022)(FSD 2021)**

SHORT QUESTIONS

1. **Differentiate between source code and object code.** **(LHR 2022) (U.B)**
2. **Give any four example of High Level Language.** **(GRW 2022) (U.B)**
3. **What do you meant by Machine Independence?** **(MTN 2022) (K.B)**
4. **Why the source code cannot be executed directly?** **(DGK 2022)(U.B)**
5. **What is Assembly language?** **(BWP 2022)(DGK 2021)(K.B)**

PROGRAM

"A well-defined set of instruction given to the computer to perform a specific task is called a **program."** A program is written in programming language. A person who writes a program is called **programmer**.



PROGRAMMING LANGUAGE

Programming languages are used to write computer programs. There are two types of programming languages. These are:

- Low Level Languages.
- High Level Languages.

Low level language:

- Low level language is near to computer language.
- **It is divided in to two broad categories:**

(i) Machine language

(ii) Assembly language

Machine language:

- It is the native language of the computer.
- The instructions are written in binary form.
- It is the only language that is directly understands by the computer.
- It is machine dependent language.

Assembly language:

- It is one step higher than machine language.
- In assembly language machine instructions are replaced with English like word called **mnemonics.**
- Easy to write and modify program in assembly language than machine language.
- Assembly language is mostly used for writing system software.

HIGH-LEVEL LANGUAGE

A type of language that is close to human language is called high-level language. Every high-level language defines a set of rules for writing programs is called its syntax. These languages are easy to learn & understand.

MULTIPLE CHOICE QUESTIONS

1. **A set of rules for writing programs is called** **(RWP2022)**

(a) Syntax	(b) Hardware
(c) Peripheral device	(d) Both (a) & (b)

2. **The statement written by the programmer called:** **(BWP 2021)**

(a) Syntax	**(b) Source Code**
(c) Object Code	(d) Keyword

3. **Which language requires no translator program?** **(DGK 2021)**

(a) Machine Language	(b) C
(c) Assembly Language	(d) C++

4. **The lowest level of programming language is** **(FSD2022)**

(a) Java	**(b) Assembly**
(c) Pascal	(d) C++

5. **Void occupy how many bytes in memory:** **(DGK 2021)**

(a) One	(b) Two
(c) Zero	(d) Four

SHORT QUESTIONS

Q1: Differentiate between source code and object code. **(LHR 2022) (U.B)**

Ans: Following are the difference between source code and object code

Source code	Object code
The program is written in any high level programming language such as C is called source code	The compiler translates the source code into object code with .obj extension

Q2: Give any four example of High Level Language. **(GRW 2022) (U.B)**

Ans: Following are the example of high level language.

(1) C/C++: It is used to write system software.

(2) FORTRON: It has best mathematical capabilities

(3) COBOL: It is ideal for writing business application

(4) JAVA: It provides strong features network programming.

Q3: What do you meant by Machine independence? **(MTN 2022)(K.B)**

Ans: It means programs written in a high level language can be executed on many different types of computers with a little modification.

Q4: Why the source code cannot be executed directly? **(DGK 2022)(U.B)**

Ans: Source code always write in high level language. High level language is not understandable by the computer directly that is why we need to convert it into the low level language (which is understandable by the computer easily) with the help of language translators.

Q5: What is assembly language? **(BWP 2022)(DGK 2021)(K.B)**

Ans: In assembly language machine language long sequences of 0s and 1s instructions are replaced with English like words known as mnemonics pronounced as Ne-monics. Assembly language is mostly used for writing system software. A language translator assembler is used to translate an assembly language programs into machine language.

LECTURE NO. 2

PREVIEW OF TODAY'S TOPICS

☞ **Developing A Program & Turbo C++**

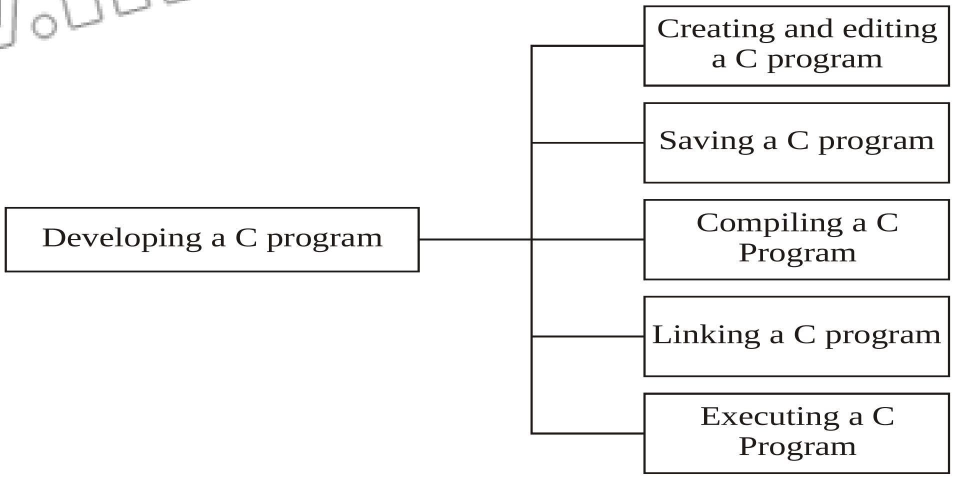
DEVELOPING A C PROGRAM & TURBO C++

LONG QUESTION

1. **Write the detail steps of writing a program in C language.** **(SWL 2022)(MTN 2022)(SGD 2018)**

SHORT QUESTIONS

1. **Write down the use of TURBO C++.** **(LHR 2021)(GRW 2022)(U.B)**
2. **Define loader.** **(LHR 2021) (K.B)**
3. **Define Linker. (Or how executable file is created?)** **(BWP 2021) (U.B)**
4. **Why does machine language execute faster?** **(SWL 2021)(U.B)**
5. **Write shortcut keys for compiling and Running in C program.** **(SWL 2021)(A.B)**



CREATING AND EDITING A C PROGRAM

- Open Turbo C++ IDE. Click on file/ new option.
- This process includes writing, modifying and deleting program statement.
- The program is written in **Turbo C++ IDE.**

SAVING A C PROGRAM

- The process of storing the program on disk is known as saving.
- C program is saved with .c extension.
- **For example,** "first.c" is a C program.
- The short cut key of saving a file is **F2.**
- The default path for saving a file is Bin folder.

COMPILING A C PROGRAM:

- The process of converting source program into object program is known as **compiling.**
- If the program is saved with .c extension, then it is known as **source program.**
- When the program is converted then is saved with .obj extension then is called **object program.**
- The short cut key for compiling a C program is **Alt+F9.**

LINKING A C PROGRAM

- The process of linking library files with object program is called known as **linking.**
- A library files must be linked with object program before the execution of the program.

- A program that combines the object program with additional library files is known as Linker.
- Linker is the part of compiler.
- When the linking is done its produce the .exe file which is called **executable file.**

EXECUTING A C PROGRAM

- The process of running executable file is known as **executing.**
- Before the running the program its first loaded into main memory.
- A program that place .exe file into main memory is called **loader.**
- The short cut key of running a program is **Ctrl+F9.**

TURBO C++

- It is a complete IDE for writing a program in C language.
- It is also called **TC Editor.**
- It is used to write, edit, compiling and also for executing a C program.
- It also provides you a powerful **debugger** that helps you to detecting and removing error in the program.

MULTIPLE CHOICE QUESTIONS

1. Which of the following key is used to save a program? **(LHR 2022)**

(a) F1 **(b) F2**
(c) F6 (d) F5

2. The TURBO C++ can compile: **(LHR 2022)(SWL 2022)(MTN 2021)**

(a) C++ programs only **(b) C & C++ Program**
(c) Turbo C programs only (d) Turbo C++ Program

3. The output of the compiler is called: **(SGD2018)**

(a) Library Code (b) Linked Code
(c) Source Code **(d) Object Code**

4. The.exe file is produced by? **(GRW2022)(SWL2019)**

(a) Assembler **(b) Linker**
(c) Debugger (d) Loader

5. C is a_________ Language: **(GRW 2022)**

(a) High Level (b) Low Level
(c) Assembly (d) Machine

SHORT QUESTIONS

1. Write down the use of TURBO C++. **(LHR2021)(GRW 2022) (U.B)**

Ans: IDE stand for integrated development environment. Turbo C++ IDE combines all the features needed to develop a C program-editor, compiler, linker, debugger, and help system into a single easy to use screen display. The IDE can be invoking by typing tc on the DOS prompt or by double clicking the TC shortcut.

2. Define loader. **(LHR 2021) (K.B)**

Ans: A program must be loaded to the memory before the execution. A loader is a program that loads an executable file from hard disk to memory where it will be executed.

3. Define Linker. (Or how executable file is created?) **(DWP 2021) (U.B)**

Ans: Linker is a program that is used to link the library files with object program and the produce the executable file, whose extension is .exe.

4. Why does machine language program execute faster? **(SWL 2021)(U.B)**

Ans: Machine language executes faster than any other language because it already has been translated into binary language (op-code). Computer does not have to translate it like any other language other than machine language.

5. Write shortcut keys for compiling and Running in C program. **(SWL 2021) (A.B)**

Ans: **Following are the shortcuts keys**

(i) The short cut key for compiling a C program is **Alt+F9.**

(ii) The short cut key of running a program is **Ctrl+F9.**

LECTURE NO. 3

PREVIEW OF TODAY'S TOPICS

☞ Basic Structure Of C Program

BASIC STRUCTURE OF C PROGRAM

LONG QUESTION

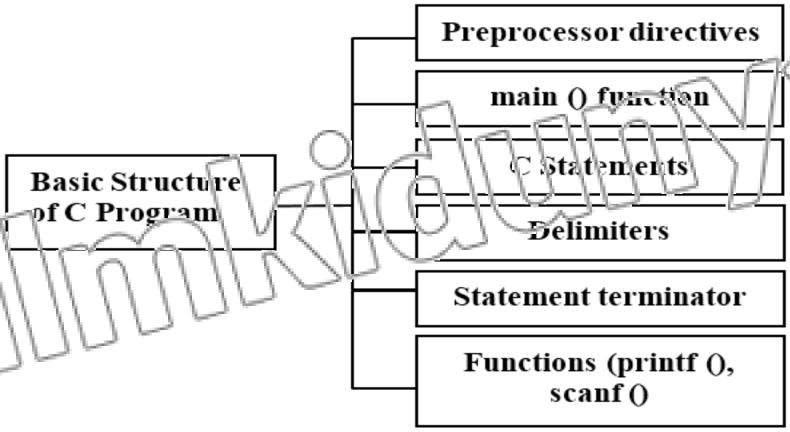
1. **What is a basic structure of C program? Discuss in detail.**

SHORT QUESTIONS

1. **Define header files.** **(DGK 2021, 2022)(SGD 2018) (K.B)**
2. **What is the purpose of # sign?** **(FSD 2021) (U.B)**
3. **Define main () function.** **(RWP 2022) (K.B)**
4. **Write down the use of delimiter?** **(SWL 2022) (BWP 2022) (FSD 2021)(U.B)**
5. **Write two uses of pre-processor directives.** **(DGK 2022) (U.B)**
6. **Define statement terminator.** **(LHR 2022) (K.B)**
7. **Write the role of define directives.** **(SWL 2022) (RWP 2022) (BWP 2022) (U.B)**
8. **Define include directive.** **(MTN 2021, 2022) (FSD 2022) (BWP 2021) (K.B)**

STRUCTURE OF C PROGRAM:

- C is a structure programming language.
- It provides a well-structured way to writing a program.
- C has very flexible structure which increases the power of the language.
- The structure is consists on three parts:
  - Preprocessor directives
  - main() function
  - C Statements
  - Delimiters
  - Statement terminator
  - Functions (printf(), scanf()



Example:

```
#include<stdio.h>
```

```
void main(void)
{
        printf("Hello I am Kipssian");
}
```

1. PREPROCESSOR DIRECTIVE

Preprocessor directive is an instruction given to the compiler before the execution of the actual program. It is the part of C++ Compiler. The **preprocessor directives** are processed by a program known as **preprocessor** it is started **with # (hash) symbol**. Semicolon is not used at the end of preprocessor directive.

There are two types of preprocessor:

(i) include preprocessor

(ii) define preprocessor

Include preprocessor:

This directive is used to enable a program to access the library. It is a key word which is written in lower case. It is used to include the header files in the program. **Header files** are the collection of standard library functions to perform different task.

Syntax:

#include<standard header files>

Example: #include<stdio.h>

#include<math.h>

Define preprocessor:

Define preprocessor is used to define a constant into the program. It can be define anywhere in the program.

Syntax: #define constant_ name expression

Example: #define PI 3.1415

#define G 10

2. main () function

- It is the place where the execution of the C program starts.
- Each program must contain main() function.
- The control always enter into the main() function
- Any number of statements can be written in the body of the main() function
- The body is enclosed in braces { } void main(void)

```
{
        Body of the program;
}
```

3. C statements:

- A statement in the C program is the actual instruction for the computer to perform a specific task.
- All the statements are written in body of the program. (in curly braces)
- Each statement is terminated **semicolon**.
- The statements of the program are used to implement the logic of the program.

Example:

printf("Hello World");

printf("This is my first program");

4. Delimiters:

The braces of main function indicate the beginning and end of the function body. These braces are called delimiters. The opening brace {indicates the beginning of a block of code while the closing brace} represents the end of a block of code.

5. **Statement Terminator:**
In a C program every statement terminates with a statement terminator i.e. semicolon(;). If any of the statement is missing the statement terminator, the compiler will report error message.

6. **Function printf() and scanf():**
printf() is used to display the program output on the screen and scanf() function is used to take input form the user.

MULTIPLE CHOICE QUESTIONS

1. **The name of header file is written in:** **(FSD 2022)**
(a) [ ] **(b) < >**
(c) ( ) (d) { }

2. **Every C program must contain:** **(MTN 2021)**
(a) printf() **(b) main()**
(c) if else if() (d) scanf()

3. **Preprocessor directives are commands for:** **(MTN 2022)**
(a) Microprocessor (b) Language processor
(c) C Preprocessor (d) Loader

4. **Define Preprocessor is used to define a:**
(a) Variable (b) Header file
(c) Label **(d) Constant**

5. **In C program which of the following is the actual instruction for the computer to perform to specific task.**
(a) Statement (b) Command
(c) Program (d) Language

SHORT QUESTIONS

1. **Define header files.** **(DGK 2021, 2022)(SGD 2018) (K.B)**

Ans: Header files are the collection of standard library functions to perform different task. There are many different header files for different purpose.

2. **What is the purpose of # sign?** **(FSD 2021) (U.B)**

Ans: This sign is used to indicate that it is an instruction for the C preprocessor.

3. **Define main() function.** **(RWP 2022) (K.B)**

Ans: The main() is the function where the execution of the C program start. Every C program has a main function. The program statements are written in the body of main function, the body is enclosed in braces { }.

```
void main (void)
{
        Body of the main function;
}
```

4. **Define statement terminator.** **(LHR 2022) (K.B)**

Ans: In a C program every statement terminates with a statement terminator i.e. semicolon(;). If any of the statement is missing the statement terminator, the compiler will report error message.

5. **Write the role of define directives.** **(SWL 2022) (RWP 2022) (BWP 2022) (U.B)**

Ans: #define directive is used to define constant macro.

The syntax of define directive is

#define Macro_Name expression

The expression may be constant, arithmetic expression or string. C preprocessor replace each occurrence of the identifier Macro_ name with the value of expression and the expression cannot be change during the program execution.

Example:

```
#define PI          3.142857
```

LECTURE NO. 4

PREVIEW OF TODAY'S TOPICS

☞ **Bugs, Debugging & Bugs Types**

BUGS, DEBUGGING & BUGS TYPES

LONG QUESTION

1. **What is debugging? Explain all different types of errors un-covered by debugging Process?** **(GRW 2022) (FSD 2022) (LHR 2021) (BWP 2021) (DGK 2021)**

SHORT QUESTIONS

1. **What is the debugging?** **(FSD 2021) (K.B)**
2. **What is syntax error?** **(MTN 2021) (K.B)**
3. **Define logical error.** **(SGD 2018) (K.B)**
4. **What do you know about Runtime error?** **(RWP 2018)(U.B)**
5. **Write the purpose of complier.** **(LHR 2021) (U.B)**
6. **What is Structured programming:** **(LHR 2021) (K.B)**
7. **Define Un-Structured programming.** **(FSD 2022) (BWP 2021) (A.B)**

BUG

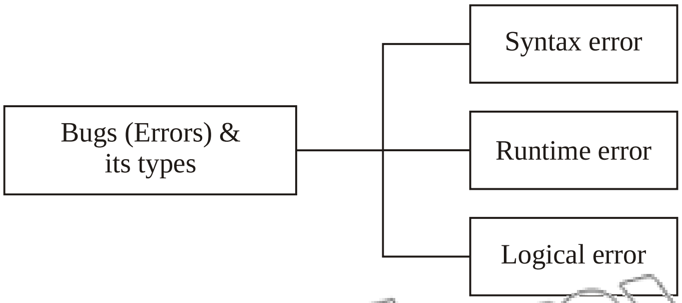
"An error in a computer program is called **bug**". The compiler detects errors and display the message

DEBUGGING:

"The process of finding and removing errors from the program is called **debugging".**

- The error must be removed from the program if you want to execute it.

BUGS TYPES:



SYNTAX ERROR:

- When a programmer violates one or more grammatical rule of program is called **syntax error.**
- **Compiler detects the syntax error.**
- A program cannot convert into machine code if it contains syntax error.

Example:

- A misspelled keyword is used in the program.
- Any of the delimiter is missing.

RUN TIME ERROR

- When the error occurs during program execution is called **runtime error.**
- It crashes the program.
- **This type of error only occurs during execution.**
- Usually this error causes by the wrong input from the user.

Example:

- Dividing a number by 0 (zero)
- Open a file, which does not exist.

LOGICAL ERROR

- An error occurs due to poor logic of the programmer.
- It is most difficult to detect because compiler cannot detects it as well as it does not crashes the program.
- **These errors can only be detected by examining the whole program thoroughly.**

Example:

- By using wrong condition.
- By using wrong formula

UNSTRUCTURED & STRUCTURED PROGRAMMING

UNSTRUCTURED PROGRAMMING:

In unstructured programming languages, the entire logic of the program is implemented in a single module or function, which makes the program error prone, difficult to understand, modify and debug**.**

STRUCTURED PROGRAMMING:

- In this type of programming the program is divided in sub program or sub function.
- In this type of programming the program is easy to write, easy to modify, and also easy to debug.
- There are also less possibility of error in the Structured programming.

MULTIPLE CHOICE QUESTIONS

1. **Debug is the process of:** **(MTN 2022)**
   (a) Creating bugs in program **(b) Identifying and removing errors**
   (c) Identifying errors (d) Removing errors
2. **Misspelled word while writing a C language program is a:**
   (a) Syntax error (b) Logical error
   (c) Runtime error (d) Both (a) & (b)
3. **Which error occur when the program directs the computer to perform an illegal operation?** **(FSD 2021)**
   (a) Runtime error (b) Logical error
   (c) Conceptual error (d) Syntax error
4. **A program's syntax errors are detected by:** **(BWP 2021)**
   (a) Compiler (b) Debugger
   (c) Linker (d) Loader
5. **Division by zero is an example of:** **(RWP 2018, 2019)**
   (a) Compile error **(b) Runtime error**

(c) Logical error (d) None

SHORT QUESTIONS

1. What is the debugging? **(FSD 2021) (K.B)**

Ans: The process of finding and removing errors in the program is called debugging. There are three types of programming errors. These are

(i) Syntax Errors
(ii) Logical Errors
(iii) Runtime Errors

2. What is syntax error? **(MTN 2021) (K.B)**

Ans: Syntax error occur when syntax of a programming language are not followed in writing the source code. The compiler detects these errors at compiling time of source code. The compiler does not compile a program that contains syntax errors. Syntax errors are easy to detect and remove.

Some examples of syntax error are given below:

- Missing any of delimiters i.e. **{** or **}**
- Incorrect spelling of any keyword.
- Missing semicolon (**;**) at the end of statement.

3. What are logical errors? **(SGD 2018) (K.B)**

Ans: Logical Errors:

The errors in the logic of the program are called logical error. The compiler cannot detect logical errors. A program with logical errors is compiled and run successfully but it does not give correct result.

The logical errors are difficult to detect. Logical errors can only be detected by examining all the statements of the program one by one. It is a very time consuming and lengthy process.

In C program there can be many causes of logical errors.

Some examples of logical errors are given below:

- The sequence of instructions used in a program may be incorrect.
- The mathematical formulas as used in program instructions may be incorrect etc.

4. What do you know about Runtime error? **(RWP 2018)(U.B)**

Ans:

The errors that occur during the execution of program are called the runtime errors. When a runtime error occurs, the computer stops the execution of program and displays an error message.

These types of errors may occur due to the following reasons.

- Dividing a number by zero
- Trying to open a file which is not created

5. **Write the purpose of compiler.** **(LHR 2021) (U.B)**

Ans: A compiler is a program that translates the source program into object program (machine code) as a whole. A program written in high level language is called source code. A program written in machine language is called object program. The compiler converts the source program written in high level or assembly language into machine code called object code. Compiler also checks and reports syntax errors in the program.

6. What is Structured programming: **(LHR 2021) (K.B)**

Ans: In this type of programming the program is divided in sub program or sub function. In this type of programming the program is easy to write, easy to modify, and also easy to debug.

7. Define Un-Structured programming. **(FSD 2022) (BWP 2021) (A.B)**

Ans: In unstructured programming languages the entire logic of the program is implemented in a single module or function, which makes the program error prone, difficult to understand, modify and debug.

STUDENTS LEARNING OBJECTIVES (SLOs)

MULTIPLE CHOICE QUESTIONS

Knowledge Based Questions

(i) The set of instructions given to computer to solve a problem is called:

(a) Hardware **(b) Program**

(c) Assembler (d) None of these

(ii) The C language was developed by:

(a) Dennis Ritchie (b) Van Neumann
(c) Dr. Howard (d) Herman Hollerith

(iii) The standard version of C language is known as:

(a) K&RC **(b) ANSI C**
(c) ITU-C (d) Turbo C++

(iv) C compiler translate source code into:

(a) Source code (b) Executable code
(c) Object code (d) None of these

(v) The program that produce .exe file. **(LHR 2022)**

(a) Linker (b) Loader
(c) Compiler (d) All of above

Understanding Based Questions

(i) C was designed to write programs for:

(a) Windows operating system (b) Solaris operating system
(c) Unix operating system (d) OS/2 operating system

(ii) The name of Header file is write between:

(a) [ ] (b) ( )
(c) <> (d) { }

(iii) The program is divided into smaller modules or functions in:

(a) Structured languages (b) Unstructured languages
(c) Both (a) & (b) (d) None of these

(iv) Missing a statement terminator cause:

(a) Runtime error (b) Logical error
(c) Syntax error (d) All of above

(v) Divide a number by zero is a:

(a) Syntax error **(b) Runtime error**
(c) Logical error (d) None of these

Application Based Questions

(i) The extension of C language program is:

(a) .cpp **(b) .c**
(c) .exe (d) .obj

(ii) The default path for saving the C program is:

(a) Include folder (b) Lib folder
(c) Bin folder (d) None of these

(iii) To run executable file press.

(a) Ctrl + F5 (b) Alt + F5
(c) Ctrl + F9 (d) None of these

(iv) Debug is the process of:

(a) Creating bugs in program **(b) Identifying and removing errors**
(c) Identifying Errors (d) Removing Errors

(v) Void occupy how many bytes in memory? **(SWL 2022) (DGK 2022) (DGK 2021)**

(a) Zero (b) One
(c) Two (d) Four

SHORT QUESTIONS

Knowledge Based Questions

Q.1 What is IDE?

Ans: IDE stands for integrated development environment. Turbo C++ IDE combines all the features needed to develop a C program- editor, compiler, linker, debugger, and help system into a single easy to use screen display. The IDE can be invoke by typing tc on the DOS prompt or by double clicking the TC shortcut.

Q.2 What the meant by linking a C program.

Ans: Linking is the process in which the object file linked to many other library files to produce executable file. This is done by the linker. To invoke linker select compile| Link from the menu bar.

Q.3 What is purpose of main function? **(RWP 2022)**

Ans: The definition of the main function comes next to the specification of the #include preprocessor directive. In fact The main() is the function where the execution of the C program start. Every C program has a main function. The program statements are written in the body of main function, the body is enclosed in braces { }.

The syntax of main function is

```
void main(void)
{
Body of the main function
}
```

Q.4 What is the purpose of statement terminator? **(LHR 2022)**

Ans: In a C program every statement terminates with a statement terminator i.e. semicolon (;). If any of the statement is missing the statement terminator, the compiler will report error message.

Q.5 What is define directive? **(SWL 2022) (MTN 2021)**

Ans: #define directive is used to define constant macro. The syntax of define directive is

#define Macro_Name expression

The expression may be constant, arithmetic expression or string. C preprocessor replace each occurrence of the identifier Macro_ name with the value of expression and the expression cannot be change during the program execution.

<u>Example:</u>

#define PI 3.142857

<u>Understanding Based Questions</u>

Q.1 How a C program does is saved?

Ans: To save a C program, select File| Save command from the menu bar or pressing F2 key a dialogue box will appear, type the name and the location in the text box and press Enter key.

Q.2 What is the difference between compiler and assembler?

Ans: Following are the difference between compiler and assembler.

Compiler	Assembler
• A compiler is program that translates the high level program into machine language as a whole. A program contain any error cannot be compiled.	• An assembler is a program that translates the assembly language program into machine language.

Q.3 Write the function keys to save and open C program.

Ans: F2 function key is used to save a C program and F3 function key is used to open an existing program.

Q.4 What do you know about C-statement?

Ans: The instructions written in C language to write program is called C statements. For example printf and scanf statements.

Q.5 What necessary steps taken to prepare a C program for execution?

Ans: Creating and editing a C program:

(i) Saving a C program

(ii) Compile a C program

(iii) Linking a C program

(iv) Executing a C program

Application Based Questions

Q.1 List two reasons why it would be preferable to write a program in C rather than machine language. **(LHR 2022)**

Or

Why would a programmer prefer to use high level language?

Ans: Every machine instruction consists of strings of 0s and 1s. So it is very difficult to remember these long sequences of 0s and 1s, therefore writing program in machine language are very difficult and error prone. The programs in C language are close to human language and far from machine language and are very easy to learn.

Q.2 Why Machine Language does execute faster? **(SWL 2021)**

Ans: Machine language executes faster than any other language because it already has been translated into binary language (op-code). Computer does not have to translate it like any other language other than machine language.

Q.3 How to create and edit C program in turbo C++?

Ans: To write the C program, open edit window by selecting File| New command from the menu bar. A double line border window appears on the screen. The cursor blink inside the window represents the starting point to write a program. We can expand this window by selecting Window| Zoom command from the menu bar or pressing F5 key. The vertical and horizontal scroll bars or arrow keys can be used to navigate in the program.

Q.4 How to executing a C language program in turbo C++?

Ans: Before executing a program, it must be loaded into memory. This is done by loader. To invoke loader select Run| Run from the menu bar or pressing Ctrl + F9 key. When a program is executed, the screen flicker and output screen will disappear. To see program output select Window| User Screen or press Alt + F5

Q.5 How pre-processor directive works?

Ans: The preprocessor directives are commands that give instructions to C preprocessor. The preprocessor examines the source code before actual compilation of code begins. A preprocessor directive always begins with the symbol (#).

Example:

#include and #define.