Module 2

VIKRAM KUMAR

**Que1 - Write an essay covering the history and evolution of C programming. Explain its importance and why it is still used today**

Ans **The history of c language**

* Dennis Ritchie is the developer of c language in 1972
* At & T, s bell lab, USA
* C language named from BCPL language
* BCPL language improves and make B language

**Importance of c programming**

* C set of built-in function
* Its access to low- level system resource
* C has become popular for embedded system
* C is highly portable

**Why it is still used**

* It is still used to its efficiency, portability and ability to connect directly with hardware.
* It is essential in system level programming and performance of critical applications.

**Que2- Describe the steps to install a C compiler (e.g., GCC) and set up an Integrated Development Environment (IDE) like DevC++, VS Code, or Code Blocks.**

Ans-

**1st install GCC compiler**

* Download MinGW
* Install and select components choose “gcc”
* Add bin folder path and set the environment variables

**2nd install the dev-C++**

1. Download dev c++
2. Run and install and complete the setup
3. Open dev c++, and set the path of GCC compilers

**3rd install c/c++ extensions**

1. Download vs code
   * + - Go to vs code website and download the installer for window
2. Install vs code
   * + - Run the downloaded installer and follow the installation instructions
3. Install c/c++ extensions
   * + - Open vs code and go to the extensions view by clicking the square icon in the activity bar on the side of the window
       - Search for c/c++ and install the extension by Microsoft.

**Que3- Explain the basic structure of a C program, including headers, main function, comments, data types, and variables. Provide examples**

Ans- #include<stdio.h> **(header file)**

Int main () **(entry point of the program where execution starts)**

**(//) = single line comment**

**(/\* content \*/) = multi-line comments**

{

Int age =10; **// integer variable**

Float height =10.1; **//float variable**

Char grades =’a’; **//character variable**

Double = **store big value;**

**(Int, float, char- define the type of data a variable can hold.)**

Variable – **containers for storing data.**

**// print values**

Printf(“age: %d \n”,age); **//%d for int**

Printf(“height: %.f \n”,height); **//%f for float**

Printf(“grade: %c \n”,grade); **//%c for char**

Return 0; **// end the program**

}

**OUTPUT –**

Age: 20

Height: 5.7

Grade: a

**Que3- Write notes explaining each type of operator in C: arithmetic, relational, logical, assignment, increment/decrement, bitwise, and conditional operators.**

Ans- arithmetic operators are use for a mathematical operation on operands

There are 5 types of if arithmetic operators

* + - * **Arithmetic operators**

1. **+** addition operator (adds two numbers of values ) **a + b**
2. **–** subtraction operator (subtracts right operand from left operand) **a – b**
3. **\*** Multiply operator ( multiply two numbers) **a\*b**
4. **/** Divide operator (divide two numbers ) **a/b**
5. **%** modules operators( return the remainder ) **a % b**
   * + - **Relation operators**
6. **<** less than
7. **>** greater than
8. **<=** less than or equal to
9. **>+** greater than or equal to
10. **==** equal to
11. **!=**  not equal to
    * + - **Logical operator**
12. **&&** logical **AND**
13. **||** logical **OR**
14. **!** logical **NOT**
    * + - **Assignment operators**
15. **=** simple assignment
16. **+=** plus, and assign
17. **-=** minus and assign
18. **\*=** Multiply and assign
19. **/=** divide and assign
20. **%=** modulus and assign
21. **&=** AND and assign
    * + - **increment/decrement**
22. **a++** post-increment
23. **++a** pre-increment

**Que4- Explain decision-making statements in C (if, else, nested if-else, switch). Provide examples of each.**

Ans-

* **if statement**
  + - * Executes a block of code if the condition is true

Ex:- **if (a>0){**

**printf(“positive number );**

**}**

* **If-else statements**
  + - * Executes one block if the condition is true otherwise executes another block

Ex:- **if (a > 0) {**

**Printf("Positive");**

**} else**

**{**

**Printf("non-positive");**

**}**

* **Nested if-else**
  + - * If or else contains another if-else.

Ex**:- if (a > 0) { if (a % 2 == 0) {**

**printf("Positive even");**

**} else {**

**printf("Positive odd");**

**}**

**} else {**

**printf("non-positive");**

**}**

* **Switch statements**
  + - * In the switch case where user choice ( choice only one case ) individual case from the more than one cases

Ex:- **switch (choice) {**

**case 1:**

**Printf("Option 1");**

**break;**

**case 2:**

**Printf("Option 2");**

**break;**

**default: Printf("Invalid choice");**

**}**

**Que5- Compare and contrast while loops, for loops, and do-while loops. Explain the scenarios in which each loop is most appropriate**

Ans-

* + - * **For loop**
      * **While loop**
      * **Do-while loop**
  + **For loop – (i=1; i<=10; i++) =**

First initializes, then condition check, the executes the body, and last the update is done

* + **While loop – while( i<10){**

**Printf(“hello world”);**

**i++;**

**}**

**Return 0;**

**}**

First Initializes, then condition checks, and then executes the body, and updating can be inside the body

* + **Do-while loop { do**

**{**

**Printf("This loop will run forever.\n");**

**} while (1);**

**return 0;**

**}**

do-while first executes the body and then the condition check is done.

**Que6- Explain the use of break, continue, and goto statements in C. Provide examples of each**

Ans**- break** - statements are used for terminating the program and exit from the program

**Continue** – skip the specific step of the loops when a condition is met but continue looping

**Goto** – goto statement is to jump to some part of code. For program can re-use or not..

**Que7- What are functions in C? Explain function declaration, definition, and how to call a function. Provide examples.**

Ans-

* **What is functions**
  + - * Function is a block of code which has some name for identification
      * Function needs to be defined only once and call it any numbers of time
      * Each function in a program must have a unique name
      * One function name in program must be main()
      * Main() function is the entry point of a c-program
* **there are three keys’ components of functions**

1. Function Declaration

* A function declaration tell the complier about a function’s name, return type and parameters (argument).

1. Function Definition

* The function definition provide the actual implementation of the function.
* This includes the return type, the function name, the parameters, and the body of the function.

1. Function Calling

* A Function call instruct to the compiler to execute the function.

* **There are 4 types of function**

1. With Return Type With Argument
2. With Return Type Without Argument
3. Without Return Type With Argument
4. Without Return Type Without Argument

**Que8- Explain the concept of arrays in C. Differentiate between one-dimensional and multi-dimensional arrays with examples.**

Ans- array is a collection of elements of the same type, which store in contiguous memory location

Arrays are useful for storing large amounts of data

* + - * **One dimensional array :-**

one-dimensional array is simple list of elements, of same type

syntax :- int Num [5] = {1,2,3,4,5};

* + - * **Multi-dimensional array :-**

A multi-dimensional array is a array of array. It is representing table of matrix

Syntax :- int Num [2][3] ={{1,2,3},{1,2,3}};

**Que9- Explain string handling functions like Strlen(), Strcpy(), strcat(), strcmp(), and Strchr(). Provide examples of when these functions are useful.**

Ans-

1. **Strlen** (string length) = is used to find the length of the string
2. **Strcpy** (string copy ) = is use for copying the string, duplicate or assign
3. **strcat** (string concatenate) = is concatenate the two string (combine two strings into one)
4. **strcmp**(string compare) = is used to compare two strings (sorting equality checks).
5. **Strchr** (string character) = find the character in string, search or parse a string
6. **Strlwr**(string lower) = is used to upper case character into lower case character
7. **Strupr**(string upper) = is used to lower case character into upper case characters
8. **Strrev**(string reverse) = is used to reverse the strings (last index of string to 0th index of string )print all characters in reverse