|  |  |  |
| --- | --- | --- |
| Logo_MEC  **LECTURE HANDOUTS**  **I / I**  **CSE** | **MUTHAYAMMAL ENGINEERING COLLEGE**  **(An Autonomous Institution)**  **(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)**  **Rasipuram - 637 408, Namakkal Dist., Tamil Nadu** | **L-1** |

**Course Name with Code : Programming For Problem Solving Techniques-19GES02**

**Course Faculty :M.SELVI**

**Unit : I- Introduction to C Programming Date of Lecture:**

|  |
| --- |
| **Topic of Lecture:**Introduction to computer software |
| **Introduction :**   * Software is a set of programs, which is designed to perform a well-defined function. * A program is a sequence of instructions written to solve a particular problem |
| **Prerequisite knowledge for Complete understanding and learning of Topic:**   1. Knowledge of computer 2. Knowledge of software |
| **Detailed content of the Lecture:**   * Computer: A programmable electronic device designed to accept data, perform prescribed mathematical and logical operations at high speed, and display the accurate results of these operations. * It can expanded as C – Common, O- Operating, M- Machine, P- Purposely, U - Used for , T-Technological ,E –Educational and R -Research   Components of Computer System  **1.1.1 Components of a Computer System**  Computer comprises of software and Hardware  **Categories - ThreeeMusketeers**  Computer – Software   * Software is a set of programs, which is designed to perform a well-defined function * A program is a sequence of instructions written to solve a particular problem   There are two types of software   1. System Software :  * The system software is a collection of programs designed to operate, control, and extend the processing capabilities of the computer itself * System software is generally prepared by the computer manufacturers * These software written in low-level languages, which interact with the hardware at a very basic level * System software serves as the interface between the hardware and the end users * Examples : Operating System, Compilers, Interpreter, Assemblers, etc.  1. Application Software  * Application software products are designed to satisfy a particular need of a particular environment * It is a collection of programs, often called a software package, which work together to accomplish user task, such as a spreadsheet package * Some examples: Payroll Software , Student Record Software , Income Tax Software and Railways Reservation Software   Difference between hardware and Software :   |  |  | | --- | --- | | **Hardware** | **Software** | | Hardware is further divided into four main categories:   * Input Devices * Output Devices * Secondary Storage Devices * CPU | Software is further divided into two main categories:   * Application Software * System Software | | Developed using electronic and other materials | Developed by using a programming language | | When damaged, it can be replaced with a new component | When damaged it can be installed once more using a backup copy | | Hardware is physical in nature and hence one can touch and see hardware | The software cannot be physically touched but still can be used and seen | | Hardware cannot be infected by Viruses | The software can be infected by Viruses | | An example of Hardware is hard drives, monitors, CPU, scanners, printers etc.. | An example of software is Windows 10, Adobe Photoshop, Google Chrome etc.. |   Difference between System Software and Application Software:   |  |  |  | | --- | --- | --- | | **Key** | **System Software** | **Application Software** | | Definition | System Software is the type of software which is the interface between application software and system | Application Software is the type of software which runs as per user request. It runs on the platform which is provide by system software | | Development Language | low level language | high level language | | Usage | System software is used for operating computer hardware | Application software is used by user to perform specific task | | Installation | Installed on the computer when operating system is installed | Application software are installed according to user’s requirements | | Dependency | System software can run independently, It provides platform for running application software | Application software can’t run independently. They can’t run without the presence of system software | | Examples | compiler, assembler, debugger, driver, etc | word processor, web browser, media player, etc. | |
| **Video Content / Details of website for further learning (if any):**  [https:](https://www.youtube.com/watch?v=enG7xaK7PfA)<https://www.youtube.com/watch?v=wvfAT2ahIcU>  https://www.centralacademy.ac.in/software-its-types |
| **Important Books/Journals for further learning including the page nos.:**  Computer Fundamentals and Programming in C - Reema Thareja: Oxford University Press, Second Edition.pp 1-5 |

**Course Faculty**

**Verified by HOD**

|  |  |  |
| --- | --- | --- |
| Logo_MEC  **LECTURE HANDOUTS**  **I / I**  **CSE** | **MUTHAYAMMAL ENGINEERING COLLEGE**  **(An Autonomous Institution)**  **(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)**  **Rasipuram - 637 408, Namakkal Dist., Tamil Nadu** | **L-2** |

**Course Name with Code : Programming For Problem Solving Techniques-19GES02**

**Course Faculty :M.SELVI**

**Unit : I- Introduction to C Programming Date of Lecture:**

|  |
| --- |
| **Topic of Lecture:**Program Design Tools: Algorithms, Flowcharts, Pseudo codes |
| **Introduction :**  Program Design tools are the tools used to develop a program. Popular tools and technique used to represent the programs are Algorithm, Flowchart, and Pseudocode |
| **Prerequisite knowledge for Complete understanding and learning of Topic:**   1. Knowledge on Software 2. Concept of Program |
| **Detailed content of the Lecture:**  Types of Computer Language   * Computer language is defined as code or syntax which is used to write programs or any specific applications * The computer language is used to communicate with computers * Three categories assembly language, machine language, and high-level language   1. Machine Language   * The Machine language is considered a low-level language * Other name -machine code or object code * Which is set of binary digits 0 and 1 * These binary digits are understood and read by a computer system * Example of machine language for the text “Hello World”.   01001000 0110101 01101100 01101100 01101111 00100000 01010111 01101111 01110010 01101100 01100100  2. Assembly Language   * Intermediate-level language for microprocessors * It is second-generation language  1. High-Level Language  * The high-level language is easy to understand and * human-readable program * Examples: C++, C, JAVA, FORTRAN, etc..   **Algorithm :**   * An algorithm in general is a sequence of steps to solve a particular problem * Algorithms are universal * The algorithm you use in C programming language is also the same algorithm you use in every other language   Qualities of a good algorithm   1. Input and output should be defined precisely 2. Each steps in algorithm should be clear and unambiguous 3. Algorithm should be most effective among many different ways to solve a problem 4. An algorithm shouldn't have computer code 5. Instead, the algorithm should be written in such a way that, it can be used in similar programming languages   Examples of Algorithms In Programming  Write an algorithm to add two numbers entered by user  Step 1: Start  Step 2: Declare variables num1, num2 and sum.  Step 3: Read values num1 and num2.  Step 4: Add num1 and num2 and assign the result to sum  sum←num1+num2  Step 5: Display sum  Step 6: Stop  **Flowchart**   * A flowchart is a type of diagram that represents a workflow or process * A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. * The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows   Some of the symbols used for flowchart and Example is shown below:    **Pseudocode**   * It cannot be compiled or run like a regular program. * Pseudocodecan be written how you want. * Syntax is a set of rules on how to use and organize statements in a programming language   Advantages of Pseudocode   * Improves the readability of any approach * It’s one of the best approaches to start implementation of an algorithm * Acts as a bridge between the program and the algorithm or flowchart * Also works as a rough documentation, so the program of one developer can be understood easily when a pseudo code is written out * In industries, the approach of documentation is essential, and that’s where a pseudo-code proves vital * The main goal of a pseudo code is to explain what exactly each line of a program should do, hence making the code construction phase easier for the programmer   Examples of Pseudocode  If student's grade is greater than or equal to 60  Print "passed"  else  Print "failed" |
| **Video Content / Details of website for further learning (if any):**  [https:](https://www.youtube.com/watch?v=enG7xaK7PfA) <https://www.youtube.com/watch?v=NXQOekzLwFA>  <https://www.edrawsoft.com/flowchart/program-flowchart-definition.html>  https://www.edureka.co/blog/introduction-to-c-programming-algorithms |
| **Important Books/Journals for further learning including the page nos.:** Computer Fundamentals and Programming in C - Reema Thareja: Oxford University Press, Second Edition.pp429-431 |

**Course Faculty**

**Verified by HOD**

|  |  |  |
| --- | --- | --- |
| Logo_MEC  **LECTURE HANDOUTS**  **I / I**  **CSE** | **MUTHAYAMMAL ENGINEERING COLLEGE**  **(An Autonomous Institution)**  **(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)**  **Rasipuram - 637 408, Namakkal Dist., Tamil Nadu** | **L-3** |

**Course Name with Code : Programming For Problem Solving Techniques-19GES02**

**Course Faculty :M.SELVI**

**Unit : I- Introduction to C Programming Date of Lecture:**

|  |
| --- |
| **Topic of Lecture:**Structure of a C program, Writing the first C program |
| **Introduction :**  It represents the way how C Program is written and what is the part present in its structure. |
| **Prerequisite knowledge for Complete understanding and learning of Topic:**   * Program Design Tools * Basic concpt of Program software |
| **Detailed content of the Lecture:**  Structure of a C program :C program basically consists of the following parts   * Comments * Preprocessor Commands * Functions * Variables * Statements & Expressions     Documentation section:   * This is a comment block, used by others to understand the code. and * It is ignored by the compiler. * Comment can be used anywhere in the program to add info about the program or code block, * Which will be helpful for developers to understand the existing code in the future easily?   // Comment - single Comment line  /\* Comment \*/ - multiple comment line  Preprocessor command   * The first line of the program    #include <stdio.h>   * is a preprocessor command, which tells a C compiler to include stdio.h file before going to actual compilation   Definition section  #define   * variable value –used to define values for the variables globally.   Main function-   * main() is the main function where the program execution begins. * It consists of Declaration part & Execution part / Body of main function   Declaration part   * Is used to declare all variables that will be used within the program.   Execution part / Body of main function   * There needs to be at least one statement in the executable part, and these two parts are declared within the opening and closing curly braces of the main().   Function   * The sub-program section deals with all user-defined functions that are called from the main() * These user-defined functions are declared and usually defined after the main() function   printf(...)   * is another function available in C which causes the message to be displayed on the screen   Writing the first C program:  #include <stdio.h> // Preprocessor command  int main() // main function  {     /\* Our first simple C basic program \*/ -// Comment line  printf("Hello World! ");    getch();     return 0;  }  OUTPUT:Hello World! |
| **Video Content / Details of website for further learning (if any):**  <https://www.youtube.com/watch?v=mLXsoqkTiiA>  https://www.studytonight.com/c/first-c-program.php |
| **Important Books/Journals for further learning including the page nos.:**  Computer Fundamentals and Programming in C - Reema Thareja: Oxford University Press, Second Edition.pp14-16 |

**Course Faculty**

**Verified by HOD**

|  |  |  |
| --- | --- | --- |
| Logo_MEC  **LECTURE HANDOUTS**  **I / I**  **CSE** | **MUTHAYAMMAL ENGINEERING COLLEGE**  **(An Autonomous Institution)**  **(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)**  **Rasipuram - 637 408, Namakkal Dist., Tamil Nadu** | **L-4** |

**Course Name with Code : Programming For Problem Solving Techniques-19GES02**

**Course Faculty :M.SELVI**

**Unit : I- Introduction to C Programming Date of Lecture:**

|  |
| --- |
| **Topic of Lecture:**Keywords, Identifiers, Basic Data Types in C |
| **Introduction :**  Characters are grouped together to form meaningful words and these meaningful words are termed as Tokens. Tokens are broadly categorized into following categories and they are as follows:Keywords, Identifiers, Constants, Operators, Special symbols |
| **Prerequisite knowledge for Complete understanding and learning of Topic:**   * Structure of a C program * Knowledge on C Tokens |
| **Detailed content of the Lecture:**  **Keywords in C**   * Keywords are predefined, reserved words used in programming * It have a special meaning * Keywords are part of the syntax and they cannot be used as an identifier   For example: int a; // Here,  int  is a keyword   |  |  |  |  | | --- | --- | --- | --- | | **Keywords in C Programming** | | | | | [auto](https://www.programiz.com/c-programming/list-all-keywords-c-language#auto) | [break](https://www.programiz.com/c-programming/list-all-keywords-c-language#break_continue) | [case](https://www.programiz.com/c-programming/list-all-keywords-c-language#switch_case_default) | [char](https://www.programiz.com/c-programming/list-all-keywords-c-language#char) | | [const](https://www.programiz.com/c-programming/list-all-keywords-c-language#const) | [continue](https://www.programiz.com/c-programming/list-all-keywords-c-language#break_continue) | [default](https://www.programiz.com/c-programming/list-all-keywords-c-language#switch_case_default) | [do](https://www.programiz.com/c-programming/list-all-keywords-c-language#do_while) | | [double](https://www.programiz.com/c-programming/list-all-keywords-c-language#double_float) | [else](https://www.programiz.com/c-programming/list-all-keywords-c-language#if_else) | [enum](https://www.programiz.com/c-programming/list-all-keywords-c-language#enum) | [extern](https://www.programiz.com/c-programming/list-all-keywords-c-language#extern) | | [float](https://www.programiz.com/c-programming/list-all-keywords-c-language#double_float) | [for](https://www.programiz.com/c-programming/list-all-keywords-c-language#for) | [goto](https://www.programiz.com/c-programming/list-all-keywords-c-language#goto) | [if](https://www.programiz.com/c-programming/list-all-keywords-c-language#if_else) | | [int](https://www.programiz.com/c-programming/list-all-keywords-c-language#int) | [long](https://www.programiz.com/c-programming/list-all-keywords-c-language#short_long_signed_unsigned) | [register](https://www.programiz.com/c-programming/list-all-keywords-c-language#register) | [return](https://www.programiz.com/c-programming/list-all-keywords-c-language#return) | | [short](https://www.programiz.com/c-programming/list-all-keywords-c-language#short_long_signed_unsigned) | [signed](https://www.programiz.com/c-programming/list-all-keywords-c-language#short_long_signed_unsigned) | [sizeof](https://www.programiz.com/c-programming/list-all-keywords-c-language#sizeof) | [static](https://www.programiz.com/c-programming/list-all-keywords-c-language#static) | | [struct](https://www.programiz.com/c-programming/list-all-keywords-c-language#struct) | [switch](https://www.programiz.com/c-programming/list-all-keywords-c-language#switch_case_default) | [typedef](https://www.programiz.com/c-programming/list-all-keywords-c-language#typedef) | [union](https://www.programiz.com/c-programming/list-all-keywords-c-language#union) | | [unsigned](https://www.programiz.com/c-programming/list-all-keywords-c-language#short_long_signed_unsigned) | [void](https://www.programiz.com/c-programming/list-all-keywords-c-language#void) | [volatile](https://www.programiz.com/c-programming/list-all-keywords-c-language#volatile) | [while](https://www.programiz.com/c-programming/list-all-keywords-c-language#do_while) |   **C Identifiers**   * Identifier refers to name given to entities such as variables, functions, structures etc. * Identifiers must be unique * They are created to give a unique name to an entity to identify it during the execution of the program called variable * To indicate the storage area, each variable should be given a unique name (identifier)   For example:  int a, b; // Here, a and b  are identifiers  Rules for naming identifiers   1. A valid identifier can have letters (both uppercase and lowercase letters), digits and underscores 2. The first letter of an identifier should be either a letter or an underscore, it can be followed by letter/digit. 3. You cannot use keywords as identifiers. 4. It should not have space in the middle.   Example  valid (Name, RollNo, A123, CSE\_A)  invalid ( 12Abc, CSE#, Roll No…)  **Data Types in C**   * A data type specifies the type of data that a variable can store such as integer * There are the following data types in C language.  |  |  | | --- | --- | | **Types** | **Data Types** | | Basic DT | int, char, float and double | | Derived DT | Array,pointer, structure and union | | Enumeration DT | enum | | Void DT | void |  |  |  |  | | --- | --- | --- | | **Data Types** | **Memory Size** | **Range** | | char | 1 byte | −128 to 127 | | signed char | 1 byte | −128 to 127 | | unsigned char | 1 byte | 0 to 255 | | short | 2 byte | −32,768 to 32,767 | | signed short | 2 byte | −32,768 to 32,767 | | unsigned short | 2 byte | 0 to 65,535 | | int | 2 byte | −32,768 to 32,767 | | signed int | 2 byte | −32,768 to 32,767 | | unsigned int | 2 byte | 0 to 65,535 | | **short int** | 2 byte | −32,768 to 32,767 | | signed short int | 2 byte | −32,768 to 32,767 | | unsigned short int | 2 byte | 0 to 65,535 | | **long int** | 4 byte | -2,147,483,648 to 2,147,483,647 | | signed long int | 4 byte | -2,147,483,648 to 2,147,483,647 | | unsigned long int | 4 byte | 0 to 4,294,967,295 | | **float** | 4 byte | 1.2E-38 to 3.4E+38 | | **double** | 8 byte | 2.3E-308 to 1.7E+308 | | **long double** | 10 byte | 3.4E-4932 to 1.1E+4932 |  * int:Integers are whole numbers that can have both zero, positive and negative values but no decimal values. Ex : int a=2; * float and double: float and double are used to hold real numbers,Ex : float a=2.345; * char: Keyword char is used for declaring character type variables.Ex: char test = 'h'; * void: void is an incomplete type. It means "nothing" or "no type“ void main ( )   **Example : Integer Output**  #include <stdio.h> int main()  {  int testInteger = 5;  printf("Number = %d", testInteger); return 0;  }  Output:Number = 5  We use %d format specifier to print int types.  Here, the %d inside the quotations will be replaced by the value of testInteger  **Example : float and double Output**  #include <stdio.h>  int main()  { float number1 = 13.5;  double number2 = 12.4;  printf("number1 = %f\n", number1);  printf("number2 = %lf", number2);  return 0; }  Output : number1 = 13.500000  number2 = 12.400000  To print float, we use %f format specifier. Similarly, we use %lf to print doublevalues.  **Example 4: Print Characters**  #include <stdio.h> int main()  {  char chr = 'a';  printf("character = %c.", chr);  return 0;  }  Output  character = a  To print char, we use %c format specifier.  **Enumeration constants**   * Keyword enum is used to define enumeration data types * Enumeration is a user defined data type in C. It is mainly used to assign names to integral constants   For example:  enum color {yellow, green, black, white};  Here, color is a variable and yellow, green, black and white are the enumeration constants having value 0, 1, 2 and 3 respectively.  **An example program to demonstrate working**  **// of enum in C**  #include<stdio.h>  enumweek{Mon, Tue, Wed, Thur, Fri, Sat, Sun};  int main( )  {  enum week day;  day = Wed;  printf("%d",day);  return 0;  }  Output: 2 |
| **Video Content / Details of website for further learning (if any):**  <https://intellipaat.com/blog/tutorial/c-tutorial/c-data-types/>  https://www.youtube.com/watch?v=HzNmyCPmJvU  https://www.geeksforgeeks.org/variables-and-keywords-in-c |
| **Important Books/Journals for further learning including the page nos.:** Computer Fundamentals and Programming in C - Reema Thareja: Oxford University Press, Second Edition.pp 19-21 |

**Course Faculty**

**Verified by HOD**

|  |  |  |
| --- | --- | --- |
| Logo_MEC  **LECTURE HANDOUTS**  **I / I**  **CSE** | **MUTHAYAMMAL ENGINEERING COLLEGE**  **(An Autonomous Institution)**  **(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)**  **Rasipuram - 637 408, Namakkal Dist., Tamil Nadu** | **L-5** |

**Course Name with Code : Programming For Problem Solving Techniques-19GES02**

**Course Faculty :M.SELVI**

**Unit : I- Introduction to C Programming Date of Lecture:**

|  |
| --- |
| **Topic of Lecture:**Variables, Constants |
| **Introduction :**  Characters are grouped together to form meaningful words and these meaningful words are termed as Tokens. Tokens are broadly categorized into following categories and they are as follows: Keywords, Identifiers, Constants, Operators, Special symbols |
| **Prerequisite knowledge for Complete understanding and learning of Topic:**   * Structure of a C program * Knowledge on C Tokens |
| **Detailed content of the Lecture:**  **Variables in C**   * A variable is a name of the memory location. * It is used to store data. * Its value can be changed, and it can be reused many times. * It is a way to represent memory location through symbol so that it can be easily identified.   syntax to declare a variable: type variable\_list;  The example of declaring the variable is given below:   1. int a; 2. float b; 3. char c; Here, a, b, c are variables. The int, float, char are the data types.  * We can also provide values while declaring the variables as given below:  1. int a=10,b=20;//declaring 2 variable of integer type 2. float f=20.8; 3. char c='A';   **Constants**   * Constants refer to fixed values that the program may not alter during its execution. * These fixed values are also called literals. * Constants can be of any of the basic data types like an integer constant, a floating constant, a character constant, or a string literal.   we can define constants in two ways as shown below:   1. Using #define preprocessor directive 2. Using a const keyword   Using #define preprocessor directive:   * This directive is used to declare an alias name for existing variable or any value. * declare a constant as shown below   #define identifierName value  identifierName: It is the name given to constant  value: This refers to any value assigned to identifierName  #include<stdio.h>  #define val 10    int main()  {      printf("Integer Constant: %d\n",val);      return 0;  }  Output: Integer Constant: 10  using a const keyword:   * Using const keyword to define constants is as simple as defining variables, the difference is you will have to precede the definition with a const keyword   #include <stdio.h>    int main()  {          const int intVal = 10;  // int constant      const float floatVal = 4.14; // Real constant      const char charVal = 'A';      // char constant          const char stringVal[10] = "ABC"; // string constant      printf("Integer constant:%d \n", intVal );      printf("Floating point constant: %.2f\n", floatVal );      printf("Character constant: %c\n", charVal );      printf("String constant: %s\n", stringVal);      return 0;  }  **Output:**  Integer constant: 10      Floating point constant: 4.14      Character constant: A     String constant:ABC |
| **Video Content / Details of website for further learning (if any):**  [https:](https://www.youtube.com/watch?v=enG7xaK7PfA)https://www.youtube.com/watch?v=bS6uNMmIoQ0  https://www.tutorialspoint.com/cprogramming/c\_constants.htm |
| **Important Books/Journals for further learning including the page nos.:** Computer Fundamentals and Programming in C - Reema Thareja: Oxford University Press, Second Edition.pp22-24 |

**Course Faculty**

**Verified by HOD**

|  |  |  |
| --- | --- | --- |
| Logo_MEC  **LECTURE HANDOUTS**  **I / I**  **CSE** | **MUTHAYAMMAL ENGINEERING COLLEGE**  **(An Autonomous Institution)**  **(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)**  **Rasipuram - 637 408, Namakkal Dist., Tamil Nadu** | **L-6** |

**Course Name with Code :Programming For Problem Solving Techniques-19GES02**

**Course Faculty :M.SELVI**

**Unit : I- Introduction to C Programming Date of Lecture:**

|  |
| --- |
| **Topic of Lecture:**Input / Output Statements in C |
| **Introduction :**  C programming language provides many built-in functions to read any given input and to display data on screen when there is a need to output the result. |
| **Prerequisite knowledge for Complete understanding and learning of Topic:**   * Structure of a C program * Knowledge on functions |
| **Detailed content of the Lecture:** Input Output (I/O) Statemens:  * scanf() -Commonly used function to take input from the user * scanf - is used when we enter data by using an input device  Syntax: scanf (“format string”, &arg1, &arg2, …..); C Output   * printf() - is one of the main output function * This function is used for displaying the output on the screen i.e the data is moved from the computer memory to the output device   Syntax: printf(“format string”, arg1, arg2, …..); Example :#include<stdio.h>void main(){ int a,b,c;printf(“Enter any two numbers: \n");scanf("%d %d", &a, &b);c = a + b;printf("The addition of two number is: %d", c);}Enter any two numbers:123The addition of two number is:15 Format Specifiers for I/O  As you can see from the above examples, we use   * %d for int * %f for float * %lf for double * %c for char   Here's a list of commonly used C data types and their format specifiers.   |  |  | | --- | --- | | Data Type | Format Specifier | | int | %d | | char | %c | | float | %f | | double | %lf | | short int | %hd | | unsigned int | %u | | long int | %li | | long long int | %lli | | unsigned long int | %lu | | unsigned long long int | %llu | | signed char | %c | | unsigned char | %c | |
| **Video Content / Details of website for further learning (if any):**  [https:](https://www.youtube.com/watch?v=enG7xaK7PfA) https://www.studytonight.com/c/c-input-output-function.php  <https://www.slideshare.net/rulza_9621/basic-input-and-output>  https://www.youtube.com/watch?v=66mubvPdaQA |
| **Important Books/Journals for further learning including the page nos.:** Computer Fundamentals and Programming in C - Reema Thareja: Oxford University Press, Second Edition.pp24-31 |

**Course Faculty**

**Verified by HOD**

|  |  |  |
| --- | --- | --- |
| Logo_MEC  **LECTURE HANDOUTS**  **I / I**  **CSE** | **MUTHAYAMMAL ENGINEERING COLLEGE**  **(An Autonomous Institution)**  **(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)**  **Rasipuram - 637 408, Namakkal Dist., Tamil Nadu** | **L-7** |

**Course Name with Code :Programming For Problem Solving Techniques -19GES02**

**Course Faculty :M.SELVI**

**Unit : I- Introduction to C Programming Date of Lecture:**

|  |
| --- |
| **Topic of Lecture:** Operators in C-Arithmetic, Relational |
| **Introduction :**  Operators can be defined as basic symbols that help us work on logical and mathematical operations |
| **Prerequisite knowledge for Complete understanding and learning of Topic:**   * Structure of a C program * Knowledge about operator |
| **Detailed content of the Lecture:**  Operators in C   * C programming has various operators to perform tasks including arithmetic, conditional and bitwise operations. * Operands are variables or expressions which are used in operators to evaluate the expression. * Combination of operands and operators form an Expression. * For instance a = b + c; denote an expression in which there are 3 operands a, b, c and two operator  + and =. * The association of expressions and keywords is called Statements.   For instance int a = b + c; denote a statement..   |  |  | | --- | --- | | **Types of Operators** | **Description** | | [Arithmetic\_operators](http://fresh2refresh.com/c/c-operators-expressions/c-arithmetic-operators/)  + - \* / % | These are used to perform mathematical calculations like addition, subtraction, multiplication, division and modulus | | [Assignment\_operators](http://fresh2refresh.com/c/c-operators-expressions/c-assignment-operators/)  = += -= \*= /= %= | These are used to assign the values for the variables in C programs. | | [Relational operators](http://fresh2refresh.com/c/c-operators-expressions/c-relational-operators/)  <><= >= == != | These operators are used to compare the value of two variables. | | [Logical operators](http://fresh2refresh.com/c/c-operators-expressions/c-logical-operators/)  && || ! | These operators are used to perform logical operations on the given two variables. | | [Bit wise operators](http://fresh2refresh.com/c/c-operators-expressions/c-bit-wise-operators/)  & | ^ ~ <<>> | These operators are used to perform bit operations on given two variables. | | [Conditional (ternary) operators](http://fresh2refresh.com/c/c-operators-expressions/c-conditional-operators/)  ? : | Conditional operators return one value if condition is true and returns another value is condition is false. | | [Increment/decrement operators](http://fresh2refresh.com/c/c-operators-expressions/c-increment-decrement-operators/)  ++ -- | These operators are used to either increase or decrease the value of the variable by one. | | [Special operators](http://fresh2refresh.com/c/c-operators-expressions/c-special-operators/)  & \* sizeof | &, \*, sizeof( ) and ternary operators. |   Arithmetic Operators :  **Assume variable A holds 10 and variable B holds 20**   |  |  |  | | --- | --- | --- | | **Operator** | **Description** | **Example** | | + | Adds two operands. | A + B = 30 | | − | Subtracts second operand from the first. | A − B = -10 | | \* | Multiplies both operands. | A \* B = 200 | | / | Divides numerator by de-numerator. | B / A = 2 | | % | Modulus Operator and remainder of after an integer division. | B % A = 0 | | ++ | Increment operator increases the integer value by one. | A++ = 11 | | -- | Decrement operator decreases the integer value by one. | B--=19 |   #include <stdio.h>  main()  { int a = 9,b = 4, c;  c = a+b; printf("a+b = %d \n",c);  c = a-b; printf("a-b = %d \n",c);  c = a\*b; printf("a\*b = %d \n",c);  c = a/b; printf("a/b = %d \n",c);  c = a%b; printf("Remainder when a divided by b = %d \n",c);  }  **Output :**  a+b = 13  a-b = 5  a\*b = 36  a/b = 2  Remainder when a divided by b=1  Relational Operators:Assume variable A holds 10 and variable B holds 20   |  |  |  | | --- | --- | --- | | Operator | Description | Example | | = = | Checks if the values of two operands are equal or not. If yes, then the condition becomes true. | (A == B) is not true. | | != | Checks if the values of two operands are equal or not. If the values are not equal, then the condition becomes true. | (A != B) is  true. | | > | Checks if the value of left operand is greater than the value of right operand. If yes, then the condition becomes true. | (A > B) is not true. | | < | Checks if the value of left operand is less than the value of right operand. If yes, then the condition becomes true. | (A < B) is true. | | >= | Checks if the value of left operand is greater than or equal to the value of right operand. If yes, then the condition becomes true. | (A >= B) is not true. | | <= | Checks if the value of left operand is less than or equal to the value of right operand. If yes, then the condition becomes true. | (A <= B) is not true. |   #include <stdio.h> int main() { int a = 5, b = 5, c = 10; printf("%d == %d is %d \n", a, b, a == b); printf("%d > %d is %d \n", a, b, a > b); printf("%d < %d is %d \n", a, b, a < c); printf("%d != %d is %d \n", a, b, a != c); printf("%d >= %d is %d \n", a, b, a >= b); printf("%d <= %d is %d \n", a, b, a <= c);  }  **Output:**  5 == 5 is 1 5 > 5 is 0 5 < 10 is 1 5 != 10 is 1 5 >= 5 is 1 5 <= 10 is 1  Bit Wise Operators in C:   * These operators are used to perform bit operations. * Decimal values are converted into binary values which are the sequence of bits and bit wise operators work on these bits. * Bit wise operators in C language are & (bitwise AND), | (bitwise OR), ~ (bitwise OR), ^ (XOR), << (left shift) and >> (right shift).   Consider x=40 and y=80. Binary form of these values are given below.  x = 00101000 (40) y=  01010000 (80)  All bit wise operations for x and y are given below.   1. x&y = 00000000 (binary) = 0 (decimal) 2. x|y = 01111000 (binary) = 120 (decimal) 3. ~x = 11010111 = -41 (decimal) 4. x^y = 01111000 (binary) = 120 (decimal) 5. x << 1 = 01010000 (binary) = 80 (decimal) 6. x >> 1 = 00010100 (binary) = 20 (decimal) |
| **Video Content / Details of website for further learning (if any):**  [https:](https://www.youtube.com/watch?v=enG7xaK7PfA) <https://www.youtube.com/watch?v=WGQRInmOBM8>  <https://www.programiz.com/c-programming/c-operators> |
| **Important Books/Journals for further learning including the page nos.:** Computer Fundamentals and Programming in C - Reema Thareja: Oxford University Press, Second Edition.pp32-35 |

**Course Faculty**

**Verified by HOD**

|  |  |  |
| --- | --- | --- |
| Logo_MEC  **LECTURE HANDOUTS**  **I / I**  **CSE** | **MUTHAYAMMAL ENGINEERING COLLEGE**  **(An Autonomous Institution)**  **(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)**  **Rasipuram - 637 408, Namakkal Dist., Tamil Nadu** | **L-8** |

**Course Name with Code :Programming For Problem Solving Techniques -19GES02**

**Course Faculty :M.SELVI**

**Unit : I- Introduction to C Programming Date of Lecture:**

|  |
| --- |
| **Topic of Lecture:**Logical, Conditional |
| **Introduction :**  Operators can be defined as basic symbols that help us work on logical and mathematical operations |
| **Prerequisite knowledge for Complete understanding and learning of Topic:**   * Structure of a C program * Knowledge about operator |
| **Detailed content of the Lecture:**  Logical Operators :   * variable A holds 1 and variable B holds 0, then  |  |  |  | | --- | --- | --- | | Operator | Description | Example | | && | Called Logical AND operator. If both the operands are non- zero, then the condition becomes true. | (A && B) is false. | | || | Called Logical OR Operator. If any of the two operands is non- zero, then the condition becomes true. | (A || B) is true. | | ! | Called Logical NOT Operator. It is used to reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make it false. | ! A =0 |   #include <stdio.h>  int main()  { int a = 5, b = 5, c = 10, result;  result = (a == b) && (c > b);  printf("(a == b) && (c > b) is %d \n", result);  result = (a == b) || (c < b);  printf("(a == b) || (c < b) is %d \n", result);  result = !(a != b);  printf("!(a == b) is %d \n", result);  }  (a == b) && (c > b) is 1  (a == b) || (c < b) is 1  !(a != b) is 1  !(a == b) is 0  Conditional operator:   * Conditional operators return one value if condition is true and returns another value is condition is false. * This operator is also called as ternary operator.   Syntax  : (Condition? true\_value: false\_value);  Example :  (A > 100  ?  0  :  1);  If the condition is true then expression1 is executed else expression2 is executed    #include <stdio.h>  main()  { int mark;  printf("Enter mark: "); scanf("%d", &mark);  puts(mark >= 40 ? "Passed" : "Failed");  }  Enter mark: 39 Failed |
| **Video Content / Details of website for further learning (if any):**  [https:](https://www.youtube.com/watch?v=enG7xaK7PfA) https://www.geeksforgeeks.org/conditional-or-ternary-operator-in-c-c  http://www.trytoprogram.com/c-programming/c-conditional-operator |
| **Important Books/Journals for further learning including the page nos.:**  Computer Fundamentals and Programming in C - Reema Thareja: Oxford University Press, Second Edition.pp35-45 |

**Course Faculty**

**Verified by HOD**

|  |  |  |
| --- | --- | --- |
| Logo_MEC  **LECTURE HANDOUTS**  **I / I**  **CSE** | **MUTHAYAMMAL ENGINEERING COLLEGE**  **(An Autonomous Institution)**  **(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)**  **Rasipuram - 637 408, Namakkal Dist., Tamil Nadu** | **L-9** |

**Course Name with Code :Programming For Problem Solving Techniques -19GES02**

**Course Faculty :M.SELVI**

**Unit : I- Introduction to C Programming Date of Lecture:**

|  |
| --- |
| **Topic of Lecture:**Type conversion and Typecasting |
| **Introduction :**  In type casting, a data type is converted into another data type by a programmer using casting operator |
| **Prerequisite knowledge for Complete understanding and learning of Topic:**   * Knowledge about data type * Precedence of data type |
| **Detailed content of the Lecture:**   * The type conversion process in C is basically converting one type of datatype to other * The conversion is done only between those datatypes wherein the conversion is possible ex – char to int (lowest data type to highest data type)   **Types of conversion:**  1. Implicit Type Conversion  2. Explicit Type Conversion  **Implicit Type Conversion**   * This type of conversion is usually performed by the compiler when necessary without any commands by the user. * Thus it is also called "Automatic Type Conversion". * The compiler usually performs this type of conversion when a particular expression contains more than one data type.   Rules...  if operand is :   1. char + int = int 2. Float+ double float =double float 3. Float+int=float   **Example 1:**  int a = 20;  double b = 20.5; a + b;  Here, first operand is int type and other is double. So, converted to double.  Therefore, the final answer is double a + b = 40.500000.  **Example 2:**  char ch='a';  int c =13;  a + c;  answer is ch + c = 97 + 13 = 110.  **Example 3:**  char ch='A';  int a =60; a \* b;  int variable, 65 + 60 = 125.  Explicit Type Conversion   * compiler for converting one data type to another instead the user explicitly defines within the program the datatype of the operands in the expression.   Syntax :(Datatype)variable\_name  **Example:**  double da = 4.5;  double db = 4.6;  double dc = 4.9;  int result = (int)da + (int)db + (int)dc;  printf("result = %d", result);  Output result is :12  **C - Type Casting**  Type casting is a way to convert a variable from one data type to another data type  Syntax :(type\_name) expression  #include <stdio.h>  main()  { int sum = 17, count = 5; double mean;  mean = (double) sum / count;  printf("Value of mean : %f\n", mean );  } Value of mean : 3.400000  Integer promotion   * If the operands still have different types, then they are converted to the type that appears highest in the following hierarchy |
| **Video Content / Details of website for further learning (if any):**  [https:](https://www.youtube.com/watch?v=enG7xaK7PfA)https://www.geeksforgeeks.org/difference-between-type-casting-and-type-conversion/  https://www.youtube.com/watch?v=t3gAwC277a8 |
| **Important Books/Journals for further learning including the page nos.:**  Computer Fundamentals and Programming in C - Reema Thareja: Oxford University Press, Second Edition.pp 46-48 |

**Course Faculty**

**Verified by HOD**