KG REDDY College of Engineering & Technology

Accredited by NAAC

A

REPORT

ON

CERTIFICATE COURSE

ADVANCED C PROGRAMMING

Organized by

(DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING)

From 20-07-2019 to 16-11-2019

For

II-B.Tech - I sem CSE

(Academic Year: 2019 -2020)

At

KG REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Affiliated to JNTUH)

Submitted by:

Dr.Basavaraj Chunchure
Associate Professor

Dr.J Srinivas
Associate professor

HOD

HEAD
DEPT. OF COMPUTER SCIENCE & ENGINEERING
K.G. REDDY COLLEGE OF ENGINEERING & TECHNOLOGY
CHILKUR (V), MOINABAD, R.R. DIST.501 504.

PRINCIPAL

Principal
KG Reddy College of Engineering & Technology
Chilkur (V), Moinabad (M).
R.R.Dist., Telangana.

Objectives of the course

The objective of the course is to bring together experts from academic institute and training institute for sharing of knowledge, expertise and experience in emerging trends related to the computer science and engineering topics.

1) To obtain in depth knowledge of C language.
2) To understand advanced features of C Programming Language.
3) Students will be provided with advanced knowledge of C programming language. Features like functions, structures, files, pointers, dynamic memory allocation, Bit operations & Pre-processors would be covered.

The course covered all topics of C-Programs, engineering system related to computer science engineering. Broad and individual topics are mentioned in syllabus but not limited. Specific tracks of the course had been taken for different session of the day.

As a result many keynote, tutorial and technical sessions have been prepared in accordance with course scope to discuss the challenges, opportunities and problems of application of computer science engineering in various fields.

OUTPUT:

This course was not only shared the knowledge among students but also tied up with expert for upcoming course. The main outputs are mentioned below:

- The expert shared his knowledge among students.
- Students learned from this course and tried to use the techniques for their project.
- Students interact with expert to gain their additional knowledge for future.
- Students found new ideas, concept, knowledge on technology, different application of methodologies from different session of course.
- It was created different domains of projects from this course for possible topic of computer science engineering.
- It helped to make industrial project.
- It helped to student for campus recruitment.
- It also helped to design C based projects for organization.

Summary of Participants

(a) Number of students attended this course: 115
(b) Number of students attended written exam:115
(c) Number of students qualified the exam: 85

KG REDDY College of Engineering & Technology

Accredited by NAAC

SUMMARY REPORT ON ADAVANCED C-PROGRAMMING

About Course

The certificate course on **Advanced C Programming** is concluded its work successfully by department of computer science and engineering (CSE) in KG ready college of Engineering and technology (KGRCET), Hyderabad, Telangana. This course is a forum to bring together students to discuss advanced concept of this course for software industry. Department has taken a new step for students to improve the quality of study through this course and become most wide scale , extensive, spectacular event in computer science engineering. The ten days course was held in two locations of the department (a) Department class room for theory class and (b) Department laboratory for practical class.

C is a procedural programming language. It was initially developed by Dennis Ritchie in the year 1972. It was mainly developed as a system programming language to write an operating system. The main features of C language include low-level access to memory, a simple set of keywords, and clean style, these features make C language suitable for system programming's like an operating system or compiler development.

This course is based on both theory and practical oriented course which is helped to student for making their expertise through C industry. The students of 2nd year 1st semester have been benefited in many ways from this course. More than 115 students have joined in this course as their own interest and completed this course. The trainer taught to students very nice with real time example and sharing his knowledge to develop technical skill in industry.

Scope of the Course

The role of Advanced C Programming is to be emphasized in computer science and engineering, to enhance and expertise for wide range of applications. It has different kind of applications as per organization and individual needed. It is Not only a fundamental but its permissive nature allows the user to manage program memory as it offers the feature of dynamic memory allocation which makes it much faster than any other language. Today, it is demand for every computer literate person is aware of the term Advanced C Programming

The course contains both theory and practical for applications as well as design methods based on C program related topics. The list of topics spans all the areas of the C programming domains. It covered significant recent developments in the field, both of a foundational and applicable character of this course. An important feature of this course is very useful in industry. The selected topics of this course helped to make project work. This permits also a rapid and broad dissemination of project.

KG REDDY College of Engineering & Technology

Accredited by NAAC

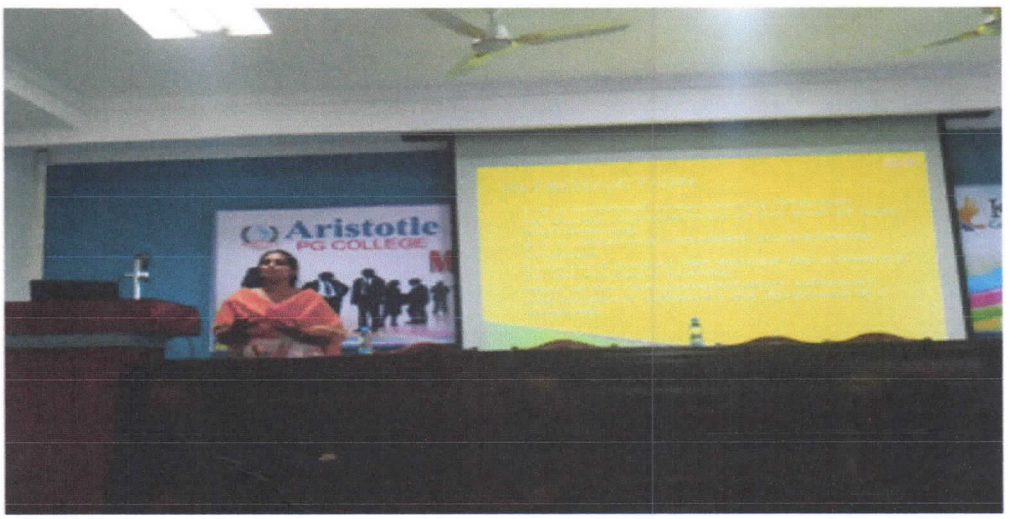
Day-1
(20-07-19)

Time: 11:00 AM to 12:00 PM

Inauguration of certificate course

The first day of certificate course started with welcoming and opening ceremony at the KGRCET conference Hall. The following dignitaries were representatives of the certificate course who were addressed and pointed out the importance on course with short welcoming speeches.

Welcome addressed by Dr. H. S. Wankhede, HOD, CSE, KGRCET
About the certificate course by Principal Dr. R. S. Jahagirdar, KGRCET
Importance of this course by expert Mrs. P.Amba Bhavani, M.V.S.R. Engg. College.Hyd.
Interaction with 2nd year 1nd semester students



Time: 12 PM to 04: 0PM

Introduction to C programming

Introduction of C:

- C is a combined programming language having the characteristics of low level an high level language.
- A C is called and embedded programming language.
- Embedded system:-We develop the software's for the hardware systems.
- Most of the Telecommunication software's and hardware software's are developed in C language.

History of C Language:

- C evolved from two previous languages, BCPL and B.
- BCPL was developed in1967 by Martin Richards as a language for writing operating systems software and compilers.
- Ken Thompson modeled may features in this language B after their counterparts in BCPL and used B in 1970 to create early versions of UNIX operating system.
- The C language was evolved from B by Dennis Ritchie at Bell Laboratiries in 1972.

Features of C Language:

- We can implement any type of application using C language. i.e, Business oriented applications and scientific related applications.
- It contains large set of operators.
- It contins large set of library functions.
- Programs in 'C' execute fast.
- It is a portable language.

Applications of C language:

- C language can be used for system softwares like-operating systems, Compilers, Interpreters, Developing Software's (printers, scanner,...)
- C language is used for developing application software like-word processor(MS word), spreadsheet(MS-Excel)
- Developing database like(Oracle, MS-Access)
- Used for Business Applications like Bank, Super Market or any Organization.
- Used for developing Scientific Applications.
- It can develop the libraries which can be used by the other programmers.

Character set of 'C' Language:

- A set of characters used in the language in order to write instructions.
- We can use all the characters from a-z (alphabets in lower case)
- We can use all the characters from A-Z (alphabets in upper case)
- We can use 0-9 digits
- We can use some special characters like <,>, , . ; : (,), “, =, + - * {, } ,~

TOKENS:

- A smallest individual unit inside the program is called the token.
- Without tokens, user cannot write statement in C-language.
- They are classified into 5 types.

1. Keywords
2. Identifiers
3. Constants
4. Datatypes
5. Operators

1. Keywords:

- Keywords are nothing but system defined identifiers.
- Keywords are reserved words of the language.
- They have specific meaning in the language and cannot be used by the programmer as variable or constant names
- C is case sensitive, it means these must be used as it is
- 32 Keywords in C Programming

2. Identifiers:

- Identifiers are user defined words.
- Identifiers are used in order to identify data(variable) or set of instructions(function name).
- In C language the identifier can be a variable name or function name.

Naming Convention or Rules to defined user defined words:

- There should not be any space between identifier name.
- Identifier cannot be keyword.
- Identifier name must start with alphabet.
- It allows only one special character that is _ (undrscore).
- The max.size of identifier is 31 characters.
- The identifier name can be given in uppercase or lowercase.

3. Constants:

- A constant is a value or an identifier whose value cannot be altered in a program. For example: 1, 2.5,
- As mentioned, an identifier also can be defined as a constant. eg. const double PI = 3.14.
- **Constants are 2 types:**

1. Numeric Constants:

Integer constants

- A integer constant is a numeric constant (associated with number) without any fractional or exponential part. There are three types of integer constants in C programming:
- decimal constant(base 10)
- octal constant(base 8)
- hexadecimal constant(base 16)

Floating-point constants

- A floating point constant is a numeric constant that has either a fractional form or an exponent form.

For example: 2.0, 0.0000234,-0.22E-5

2. Character Constants:

It classified into two types.

a)Single Character constants

b)String constants

Single Character constants

- A character constant is a constant which uses single quotation around characters. For example: 'a', 'l', 'm', 'F'

String constants

- String constants are the constants which are enclosed in a pair of double-quote marks. For example: "good" ,"x","Earth is round\n"

3. Datatypes:

- A datatypes is used for allocating memory/reserving memory for data used inside the program.
- Datatype define two character strings.

i) Type ii) Size

- Primitive Datatypes:
  - Primitive Types in ANSI C (C89)/ISO C (C90) - char, short, int, float and double.
- Integer Datatype:
- Floating or Real Data Types
- Character Datatype

Day-2
(27-07-19)

Conditionals statements : The `if` statement,The `?` operator,The `switch` statement
Exercises

In a 'C' program are executed sequentially. This happens when there is no condition around the statements. If you put some condition for a block of statements the flow of execution might change based on the result evaluated by the condition. This process is referred to as decision making in 'C.' The decision-making statements are also called as control statements.
In 'C' programming conditional statements are possible with the help of the following two constructs:
1. If statement
It is also called as branching as a program decides which statement to execute based on the result of the evaluated condition.

If statement:

It is one of the powerful conditional statement. If statement is responsible for modifying the flow of execution of a program. If statement is always used with a condition. The condition is evaluated first before executing any statement inside the body of If. The syntax for if statement is as follows:

```
if (condition)

    instruction;
```

The condition evaluates to either true or false. True is always a non-zero value, and false is a value that contains zero. Instructions can be a single instruction or a code block enclosed by curly braces { }.

The conditional operator: is kind of similar to the if-else statement as it does follow the same algorithm as of if-else statement but the conditional operator takes less space and helps to write the if-else statements in the shortest way possible.

Example : variable = Expression1 ? Expression2 : Expression3

A switch statement: Tests the value of a variable and compares it with multiple cases. Once the case match is found, a block of statements associated with that particular case is executed. Each case in a block of a switch has a different name/number which is referred to as an identifier. The value provided by the user is compared with all the cases inside the switch block until the match is found. If a case match is found, then the default statement is executed, and the control goes out of the switch block.
Example: **switch(expression)**

```
{
        case value-1:
                        Block-1;
                        Break;
        case value-2:
                        Block-2;
                        Break;
        case value-n:
                        Block-n;
                        Break;
        default:
                        Block-1;
                        Break;
}
Statement-x;
```

- The expression can be integer expression or a character expression.
- Value-1, 2, n are case labels which are used to identify each case individually. Remember that case labels should not be same as it may create a problem while executing a program. Suppose we have two cases with the same label as '1'. Then while executing the program, the case that appears first will be executed even though you want the program to execute a second case. This creates problems in the program and does not provide the desired output.
- Case labels always end with a colon (:). Each of these cases is associated with a block.
- A block is nothing but multiple statements which are grouped for a particular case.
- Whenever the switch is executed, the value of test-expression is compared with all the cases which we have defined inside the switch. Suppose the test expression contains value 4. This value is compared with all the cases until case whose label four is found in the program. As soon as a case is found the block of statements associated with that particular case is executed and control goes out of the switch.
- The break keyword in each case indicates the end of a particular case. If we do not put the break in each case then even though the specific case is executed, the switch will continue to execute all the cases until the end is reached. This should not happen; hence we always have to put break keyword in each case. Break will terminate the case once it is executed and the control will fall out of the switch.
- The default case is an optional one. Whenever the value of test-expression is not matched with any of the cases inside the switch, then the default will be executed. Otherwise, it is not necessary to write default in the switch.

Day-3
(03-08-19)

Looping and Iteration: The `for` statement, The `while` statement The `do-while` statement, `break` and `continue`, Exercises

In looping, a program executes the sequence of statements many times until the stated condition becomes false. A loop consists of two parts, a body of a loop and a control statement. The control statement is a combination of some conditions that direct the body of the loop to execute until the specified condition becomes false.

For loop:
A for loop is a more efficient loop structure in 'C' programming. The general structure of for loop is as follows:

```
for (initial value; condition; incrementation or decrementation )
{
 statements;
}
```

- The initial value of the for loop is performed only once.
- The condition is a Boolean expression that tests and compares the counter to a fixed value after each iteration, stopping the for loop when false is returned.
- The incrementation/decrementation increases (or decreases) the counter by a set value.

While Loop:
A while loop is the most straightforward looping structure. The basic format of while loop is as follows:

```
while (condition) {
        statements;
}
```

It is an entry-controlled loop. In while loop, a condition is evaluated before processing a body of the loop. If a condition is true then and only then the body of a loop is executed. After the body of a loop is executed then control again goes back at the beginning, and the condition is checked if it is true, the same process is executed until the condition becomes false. Once the condition becomes false, the control goes out of the loop.
After exiting the loop, the control goes to the statements which are immediately after the loop. The body of a loop can contain more than one statement. If it contains only one statement, then the curly braces are not compulsory. It is a good practice though to use the curly braces even we have a single statement in the body.
In while loop, if the condition is not true, then the body of a loop will not be executed, not even once. It is different in do while loop which we will see shortly.

Do-While loop:
A do-while loop is similar to the while loop except that the condition is always executed after the body of a loop. It is also called an exit-controlled loop.
The basic format of while loop is as follows:

```
do {
 statements
} while (expression);
```

int Arr [ ] = { 2, 3, 5, 4, 6, 7 };

Since, the indexing of an array starts from 0, if the 4th element needs to be accessed, Arr [3] will give you the value of the 4th element. Similarly, nth element can be accessed by Arr[n-1].

Multidimensional array: It can be considered as an array of arrays. The most commonly used multidimensional array is 2-dimensional array. It stores the elements using 2 indices, which give the information, in which row and which column a particular element is stored. A 2D array is essentially a matrix.

Declaration:

char A[3] [2] ;

Here, A is a 2D array of character data type. The array contains 3 rows and 2 columns.

This array can be used to store 3 words of length 2, for example - PI, OM, GD. These words

Here rows and columns start with 0 index. If you have to access 'M' character, you can access it by A[1] [1].

Strings:

A string is essentially a sequence of characters. In other words, we can say that string is an array of character data type. An instance of a string is called a string literal . They are enclosed in double quotes ("") . For example "HackerEarth" is a called a string literal. There are two types of string in C++: C-style string String class type

C-style strings are one dimensional array of characters. However they have a special property, they end with a null character ('\0') to signify the end of the string.

Declaration:

char str[6];

Initialization: A character array can be initialized like any other normal array

char str[6] = { 'H', 'E', 'L', 'L', 'O' };

or it can directly be initialized using a string literal

char str[ ] = "HELLO";

str is a string which has an instance (string literal) "HELLO". In spite of only 5 characters, the size of the string is 6, because the null character represents the end of the string. The null character is automatically added by the compiler as long as the size of the array is at least one larger than the size of the string. It is not mandatory that the array size (N) be exactly greater than the size of the string (n) by 1 i.e. N >= n + 1

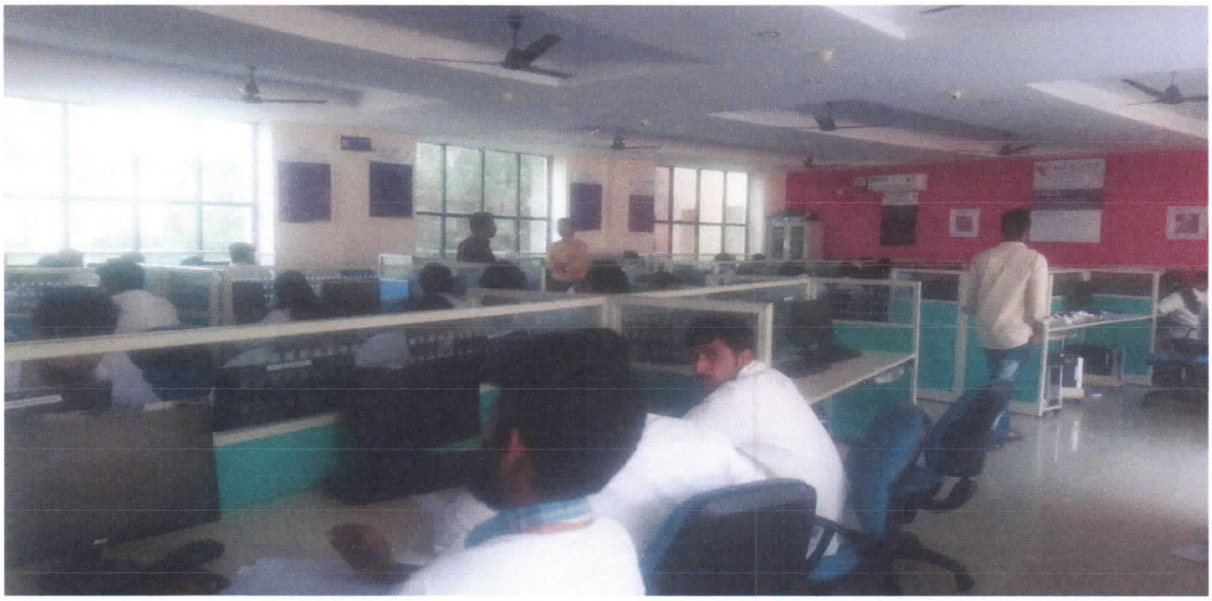
To take C-style string as an input you can simple use scanf() with '%s' format specifier or we can also use cin. It will take input until we press a space or enter. For example:

char a[10];

scanf("%s", a);

Similarly you can print the C-style string with the printf() with '%s' format specifier or cout.

printf("%s\n", a);



Day-5
(07-09-19)

Functions: `void` functions**,** Functions and Arrays**,**Function Prototyping

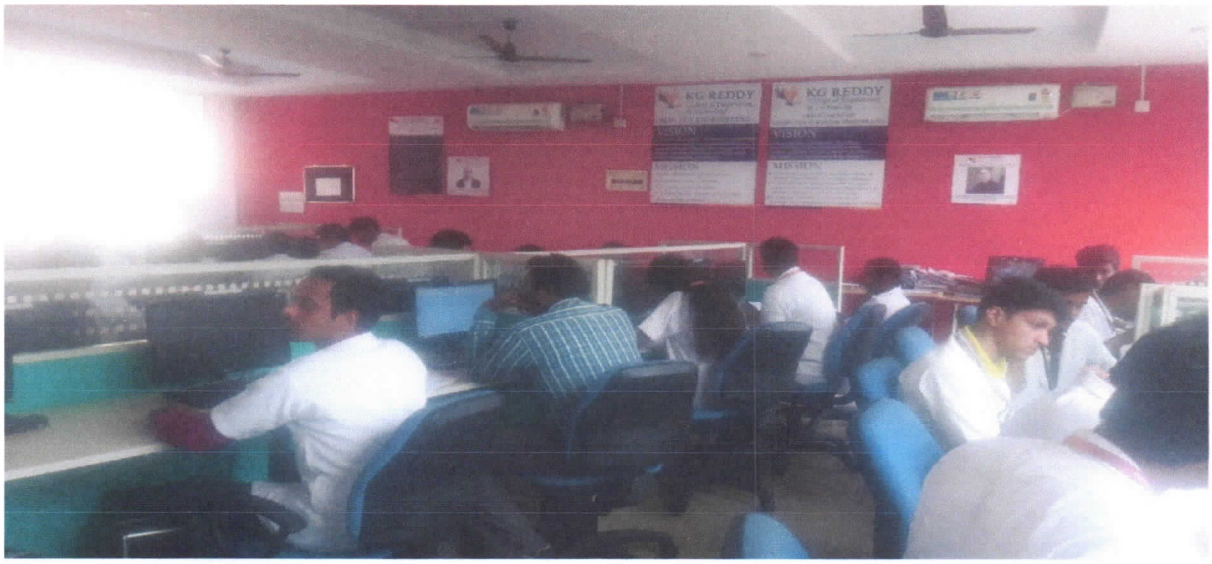
Exercises

A function: is a set of statements that take inputs, do some specific computation and produces output.The idea is to put some commonly or repeatedly done task together and make a function so that instead of writing the same code again and again for different inputs, we can call the function.

Void functions: also called nonvalue-returning functions, are used just like value-returning functions except void return types do not return a value when the function is executed. The void function accomplishes its task and then returns control to the caller. The void function call is a stand-alone statement

Functions and Arrays: C programming, creating an array for use inside a function works just like creating an array for use inside the main() function: The array is declared, it's initialized, and its elements are used. You can also pass arrays to and from functions, where the array's elements can be accessed or manipulated.

A function prototype: is simply the declaration of a function that specifies function's name, parameters and return type. It doesn't contain function body. A function prototype gives information to the compiler that the function may later be used in the program



Day-6
(21-09-19)

Further Data Types: Structures**:** Defining New Data Types**,** Unions, Coercion or Type-Casting, Enumerated Types, Static Variables, Exercises

Structure: is a user-defined datatype in C language which allows us to combine data of different types together. Structure helps to construct a complex data type which is more meaningful. It is somewhat similar to an Array, but an array holds data of similar type only In structure, data is stored in form of records.

Union is a data type in C programming that allows different data types to be stored in the same memory locations. Union provides an efficient way of reusing the memory location, as only one of its members can be accessed at a time. A union is used almost in the same way you would declare and use a structure

Type Casting - C Programming. Type casting refers to changing an variable of one data type into another. The compiler will automatically change one type of data into another if it makes sense. ... Casting allows you to make this type conversion explicit, or to force it when it wouldn't normally happen.

Enumeration type :(also called enum) is a data type that consists of integral constants. To define enums, the enum keyword is used.:#include <stdio. h>,enum suit

Static :is used with global variables and functions to set their scope to the containing file. In local variables, static is used to store the variable in the statically allocated memory instead of the automatically allocated memory

Day-7
(19-10-19)

Pointers: What is a Pointer?, Pointer and Functions, Pointers and Arrays ,Arrays of Pointers, Multidimensional arrays and pointers, Static Initialization of Pointer Arrays, Pointers and Structures, Common Pointer Pitfalls: Not assigning a pointer to memory address before using it, Illegal indirection, Exercise

Pointer is a programming language object that stores the memory address of another value located in computer memory. A pointer references a location in memory, and obtaining the value stored at that location is known as dereferencing the pointer.

Pointers give greatly possibilities to 'C' functions which we are limited to return one value. With pointer parameters, our functions now can process actual data rather than a copy of data. In order to modify the actual values of variables, the calling statement passes addresses to pointer parameters in a function

Pointer and Arrays in C. When an array is declared, compiler allocates sufficient amount of memory to contain all the elements of the array. Base address i.e address of the first element of the array is also allocated by the compiler. ... We can also declare a pointer of type int to point to the arr

Array of pointers can be used to point to an array of data items with each element of the pointer array pointing to an element of the data array. Data items can be accessed either directly in the data array, or indirectly by dereferencing the elements of the pointer array.

Static Arrays in C. This feature can be used with arrays as well. A static array has the following characteristics: ... If a static array is not explicitly initialized, its elements are initialized with the default value which is zero for arithmetic types (int, float, char) and NULL for pointers.

Pointer to a Structure in C. We have already learned that a pointer is a variable which points to the address of another variable of any data type like int , char , float etc. Similarly, we can have a pointer to structures, where a pointer variable can point to the address of a structure variable.

Not assigning: Before we learn pointers, let's learn

About addresses in C programming. Address in C. I f you have a variable var in your program, &var will give you its address in the memory. We have ... You cannot and should not do something like *pc = &c.

However **pointers** and arrays are different: A **pointer** is a variable. We can do. pa = a and pa++. An Array is not a variable. a = pa and a++ ARE **ILLEGAL**.

Day-8
(26-10-2019)

Dynamic Memory Allocation and Dynamic Structures: Malloc, Sizeof, and Free, Calloc and Realloc, Linked Lists Full Program: `queue.c`, Exercises
Advanced Pointer Topics: Pointers to Pointers, Command line inputPointers to a Function, Exercises.
Since C is a structured language, it has some fixed rules for programming. One of it includes changing the size of an array. An array is collection of items stored at continuous memory locations.

C provides some functions to achieve these tasks. There are 4 library functions provided by C defined under **<stdlib.h>** header file to facilitate dynamic memory allocation in C programming. They are:

1. malloc()
2. calloc()
3. free()
4. realloc()

"malloc" or **"memory allocation"** method in C is used to dynamically allocate a single large block of memory with the specified size. It returns a pointer of type void which can be cast into a pointer of any form.
"calloc" or **"contiguous allocation"** method in C is used to dynamically allocate the specified number of blocks of memory of the specified type. It initializes each block with a default value '0'.
"free" method in C is used to dynamically **de-allocate** the memory. The memory allocated using functions malloc() and calloc() is not de-allocated on their own. Hence the free() method is used, whenever the dynamic memory allocation takes place. It helps to reduce wastage of memory by freeing it.
"realloc" or **"re-allocation"** method in C is used to dynamically change the memory allocation of a previously allocated memory. In other words, if the memory previously allocated with the help of malloc or calloc is insufficient, realloc can be used to **dynamically re-allocate memory**.
-Example of que.c
Command line arguments :are passed to the main() method. Here argc counts the number of arguments on the command line and argv[ ] is a pointer array which holds pointers of type char which points to the arguments passed to the program.

Day-9

(02-11-2019**)**

Low Level Operators and Bit Fields: Bitwise Operators**,** Bit Fields: Bit Fields: Practical Example, A note of caution: Portability, Exercises

Many programs (***e.g.*** systems type applications) must actually operate at a low level where individual bytes must be operated on.

The ***bitwise*** operators of C a summarised in the following table:

Bitwise operators	
&	AND
|	OR
^	XOR
~	One's Compliment
	0 → 1
	1 → 0
<<	Left shift
>>	Right Shift

Portability:

Bit fields are a convenient way to express many difficult operations. However, bit fields do suffer from a lack of portability between platforms:

- integers may be signed or unsigned
- Many compilers limit the maximum number of bits in the bit field to the size of an integer which may be either 16-bit or 32-bit varieties.
- Some bit field members are stored left to right others are stored right to left in memory.
- If bit fields too large, next bit field may be stored consecutively in memory (overlapping the boundary between memory locations) or in the next word of memory.

If portability of code is a premium you can use bit shifting and masking to achieve the same results but not as easy to express or read

Day-10

(16-11-2019**)**

The C Preprocessor:#define, #undef, #include, #if -- Conditional inclusion**,** Preprocessor Compiler Control**,** Other Preprocessor Commands**,** Exercises

A Preprocessor: is a system software (a computer program that is designed to run on computer's hardware and application programs). It performs preprocessing of the High Level

Language(HLL). Preprocessing is the first step of the language processing system. Language processing system translates the high level language to machine level language or absolute machine code(i.e. to the form that can be understood by machine).
The preprocessor doesn't know about the scope rules of C. Preprocessor directives like #define come into effect as soon as they are seen and remain in effect until the end of the file that contains them; the program's block structure is irrelevant.

1.Removing comments : It removes all the comments. A comment is written only for the humans to understand the code. So, it is obvious that they are of no use to a machine. So, preprocessor removes all of them as they are not required in the execution and won't be executed as well.

2.File inclusion : Including all the files from library that our program needs. In HLL we write #include which is a directive for the preprocessor that tells it to include the contents of the library file specified. For example, #include will tell the preprocessor to include all the contents in the library file stdio.h. This can also be written using double quotes – #include "stdio.h"

3. Macro expansion : Macros can be called as small functions that are not as overhead to process. If we have to write a function (having a small definition) that needs to be called recursively (again and again), then we should prefer a macro over a function. So, defining these macros is done by pre processor.

****

KG REDDY
College of Engineering
& Technology

Ref No: KGRCET/CSE/2019-20/ 005

Date: 16/07/2019

CIRCULAR

All the students of II-Year I-semester B.Tech CSE are here by informed to enroll for the certification course on "Advanced C Programming", which is offered by KG Reddy college of Engineering and Technology from 20/07/2019 to 16/11/2019. Hence students are instructed to complete their registration on or before 19/07/2019.

HOD

HEAD
DEPT. OF COMPUTER SCIENCE & ENGINEERING
KG REDDY COLLEGE OF ENGINEERING & TECHNOLOGY
CHILKUR (V), MOINABAD, R.R. DIST.501 504.

Principal

Principal
KG Reddy College of Engineering & Technology
Chilkur (V), Moinabad (M).
R.R.Dist., Telangana.

Copy to:

1.Exam section

2. Library

3. class room

KG REDDY College of Engineering & Technology

Accredited by NAAC

KG Reddy College of Engineering & Technology
Chilkur, Moinabad, RR District
Department of Computer Science and Engineering

Certificate course, **SYLLABUS** KGRCET, Hyderabad

KGRCC0506: **Advanced C Programming**

B.Tech., CSE, II Year-I sem L-30

Course objectives:

1) To understand the basic concepts in advanced C Programming Language.
2) To learn how to write modular and readable C Programs.
3) To learn to write programs in C to solve problems using functions, pointers ,bit operators, preprocessors.

Course Outcomes:

The students would be able:

1) To obtain in depth knowledge of C language.
2) To understand advanced features of C Programming Language.
3) Design effectively the required programming components that efficiently solve computing problems.

Module -1

C Basics [3 Hrs]

a. History of C
b. Characteristics of C
c. C Program Structure
d. Variables
    i. Defining Global Variables
    ii. Printing Out and Inputting Variables
e. Constants
f. Arithmetic Operations
g. Comparison Operators
h. Logical Operators
i. Tokens
j. Data types
k. Control String
l. Exercises

Module-2 [3 Hrs]

Conditionals:

a. The `if` statement
b. The ? operator
c. The `switch` statement
d. Exercises

Module -3 [3 Hrs]

Looping and Iteration:

a. The `for` statement
b. The `while` statement
c. The `do-while` statement
d. `break` and `continue`
e. Exercises

Module -4 [3 Hrs]

Arrays and Strings

a. Single and Multi-dimensional Arrays
b. Strings
c. Exercises

Module -5 [3 Hrs]

Functions

a. `void` functions
b. Functions and Arrays
c. Function Prototyping
d. Exercises

Module -6 [3 Hrs]

Further Data Types

a. Structures
    i. Defining New Data Types
b. Unions
c. Coercion or Type-Casting
d. Enumerated Types
e. Static Variables
f. Exercises

Module -7 [3 Hrs]

Pointers

a. What is a Pointer?
b. Pointer and Functions
c. Pointers and Arrays
d. Arrays of Pointers
e. Multidimensional arrays and pointers
f. Static Initialization of Pointer Arrays
g. Pointers and Structures
h. Common Pointer Pitfalls
    i. Not assigning a pointer to memory address before using it
    ii. Illegal indirection
i. Exercise

KG REDDY
College of Engineering
& Technology

Module -8 [3 Hrs]

Dynamic Memory Allocation and Dynamic Structures

a. Malloc, Sizeof, and Free
b. Calloc and Realloc
c. Linked Lists
d. Full Program: `queue.c`

Advanced Pointer Topics

j. Pointers to Pointers
k. Command line input
l. Pointers to a Function
m. Exercises

Module -9 [3 Hrs]

Low Level Operators and Bit Fields

a. Bitwise Operators
b. Bit Fields
   i. Bit Fields: Practical Example
   ii. A note of caution: Portability

Exercises

Module -10 [3 Hrs]

The C Preprocessor

a. #define
b. #undef
c. #include
d. #if -- Conditional inclusion
e. Preprocessor Compiler Control
f. Other Preprocessor Commands
g. Exercises

Text books and References

Text Books	
1.	Programming In C (Second Edition) Publication : Pearson Education by Ashok N. Kamthane
Reference Books	
1.	Simplifying C (First Edition 2010) Publication : Dreamtech by Harshal Arolkar and Sonal Jain
2.	Programming in ANSI C (Fifth Edition 2011) Publication : Mc Graw Hill by Balagurusamy
3.	Programming in C (First Edition 2011) Publication : Oxford Higher Education by Reema Thareja

Signature of Chairman, BOS, CSE

KG REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

Chilkur (Vill) Moinabad (Mdl) R R Dist

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE COURSE ON ADVANCED C PROGRAMMING

II B Tech CSE(A & B)

Schedule

Day	Date	Topics to cover
1	20-07-2019	**C Basics:** History of C, Characteristics of C, C Program Structure Variables: Defining Global Variables, Printing Out and Inputting Variables. Constants, Arithmetic Operations, Comparison Operators Logical Operators, Tokens, Data types, Control String, Exercises
2	27-07-2019	**Conditionals:**The `if` statement,The `?` operator,The `switch` statement Exercises
3	03-08-2019	**Looping and Iteration:** The `for` statement, The `while` statement The `do-while` statement, `break` and `continue`, Exercises
4	31-08-2019	**Arrays and Strings:**Single and Multi-dimensional Arrays, Strings Exercises
5	07-09-2019	**Functions:**`void` functions, Functions and Arrays,Function Prototyping Exercises
6	21-09-2019	**Further Data Types:** Structures: Defining New Data Types, Unions Coercion or Type-Casting, Enumerated Types, Static Variables, Exercises
7	19-10-2019	**Pointers:** What is a Pointer?, Pointer and Functions, Pointers and Arrays ,Arrays of Pointers, Multidimensional arrays and pointers, Static Initialization of Pointer Arrays, Pointers and Structures, Common Pointer Pitfalls: Not assigning a pointer to memory address before using it, Illegal indirection, Exercise
8	26-10-2019	**Dynamic Memory Allocation and Dynamic Structures:** Malloc, Sizeof, and Free, Calloc and Realloc, Linked Lists Full Program: `queue.c`, Exercises **Advanced Pointer Topics:** Pointers to Pointers, Command line inputPointers to a Function, Exercises
9	02-11-2019	**Low Level Operators and Bit Fields:** Bitwise Operators, Bit Fields: Bit Fields: Practical Example, A note of caution: Portability, Exercises
10	16-11-2019	**The C Preprocessor:**#define, #undef, #include, #if -- Conditional inclusion, Preprocessor Compiler Control, Other Preprocessor Commands, Exercises

Note: Each day and date: 3hrs for A & B section

Coordinators

HOD

HEAD
DEPT. OF COMPUTER SCIENCE & ENGINEERING
K.G. REDDY COLLEGE OF ENGINEERING & TECHNOLOGY
CHILKUR (V), MOINABAD, R.R. DIST.501 504.

KG REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

Chilkur (Vill) Moinabad (Mdl) R R Dist

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE COURSE ON ADVANCED C PROGRAMMING

ATTENDANCE SHEET

YEAR: II SEM: I SEC:A DATE: 20/07/2019 TO 16/11/2019

			1	2	3	4	5	6	7	8	9	10
S.NO	**ROLL NO**	**NAME**	20/7	27/7	3/8	31/8	7/9	21/9	19/10	26/10	2/11	16/11
1	18QM1A0501	A AMARSAI REDDY										
2	18QM1A0503	A MARY SHYLU										
3	18QM1A0504	AKARAPU MANIRAJU										
4	18QM1A0505	AKASH MAKANE										
5	18QM1A0506	ALLURI ASHISH VARDHAN										
6	18QM1A0507	AMULA PAVAN KUMAR										
7	18QM1A0508	APPALA RAVI TEJA										
8	18QM1A0511	ASHWINI NAVEEN KUMAR										
9	18QM1A0512	BADHAGOWNI SHAILAJA										
10	18QM1A0513	BASHETTY VINEELA										
11	18QM1A0514	BHOSLE PRIYANKA										
12	18QM1A0515	BODA THIRUMALA HRUDAY										
13	18QM1A0516	BOJA YASHODA KRISHNA										
14	18QM1A0517	BOKKA SRINIVAS REDDY										
15	18QM1A0518	CHENGARAPU DEEPTHI										
16	18QM1A0519	CHILAM ANUCHARITHA										
17	18QM1A0520	CHINMAY MANAS										
18	18QM1A0521	DURVASULA ABHINAV										
19	18QM1A0522	DUSA SHIVA TEJA										
20	18QM1A0523	EEDAMMAGARI SIREESHA										

21	18QM1A0524	FIRAS RASHEED										
22	18QM1A0526	G MEGHANA REDDY										
23	18QM1A0527	G S D PRANAY KUMAR RAO										
24	18QM1A0528	GADDAM JAYANTH REDDY										
25	18QM1A0530	GADDAMEEDI SANDYA										
26	18QM1A0531	GANDLA ANUSHA										
27	18QM1A0532	GANGIDI HARATHI										
28	18QM1A0533	GEEDA KEERTHI REDDY										
29	18QM1A0534	GODHA RAJITHA										
30	18QM1A0535	GOGIKAR NIKHIL										
31	18QM1A0536	GOLLA SRINIVAS YADAV										
32	18QM1A0537	GONGATI ARUN KUMAR REDDY										
33	18QM1A0538	GOPULARAM AKHILESH										
34	18QM1A0539	GUDISE ANUSHA										
35	18QM1A0541	GUNDABATHULA MEGHANA SHARON										
36	18QM1A0542	GUTTULA LOKESH										
37	18QM1A0544	JALAGARI SAIKRISHNA										
38	18QM1A0545	JOGA KALYAN										
39	18QM1A0549	KAUSHIK TIWARI										
40	18QM1A0550	KAVALI BHAVANI										
41	18QM1A0551	KAVALI PRANEETH BABU										
42	18QM1A0552	KESARI MADHURI										
43	18QM1A0553	KESEPALLY TARUN REDDY										
44	18QM1A0554	KESHAPALLY TARUNI										
45	18QM1A0555	KOPPISETTI HARANI										
46	18QM1A0556	KUKUTLA SAINISHANTH REDDY										
47	18QM1A0557	KUMMARI KAVYA SREE										
48	18QM1A0559	MADABUSHI SAKETH										

49	18QM1A0560	MADDALA SRI JANA SAI PAVAN	Pavan	Pavan	Pavan	Pavan	Pavan	Pavan	Pavan	Pavan	Pavan	Pavan	
50	16QN1A0521	D. Srinath kumar.	Srinath	Srinath	Srinath	Srinath	Srinath	Srinath	Srinath	Srinath	Srinath	Srinath	

HOD

KG REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

Chilkur (Vill) Moinabad (Mdl) R R Dist

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE COURSE ON ADVANCED C PROGRAMMING

ATTENDANCE SHEET

YEAR: II **SEM: I** **SEC:B** **DATE: 20/07/2019 TO 16/11/2019**

			1	2	3	4	5	6	7	8	9	10
S.NO	**ROLL NO**	**NAME**	20/7	27/7	3/8	31/8	7/9	21/9	19/10	26/10	2/11	16/11
1	17QM1A0524	DAMERA VIJAY										
2	17QM1A0531	GADHIRAJU LAKSHMI SURAJ										
3	17QM1A0548	KAREDLA NISHWANTH REDDY										
4	17QM1A0584	SUGURI NIKHIL SAGAR										
5	17QM1A05A9	GANAPA SHIVANATH REDDY										
6	18QM1A0561	MADDURI MAMATHA										
7	18QM1A0562	MALGARI RAVALI REDDY										
8	18QM1A0564	METTLUKUNTA AKHILA										
9	18QM1A0566	MOHAMMED SUHAIBUDDIN										
10	18QM1A0567	MOKILA ANUSHA										
11	18QM1A0568	MUNGI MADHAN MOHAN REDDY										
12	18QM1A0569	NIMMALA RAJESHWARI										
13	18QM1A0571	P MADHUSUDHAN REDDY										
14	18QM1A0572	P MUKESH YADAV										
15	18QM1A0573	P RAKESH REDDY										
16	18QM1A0574	PASHAMONI LAVANYA										
17	18QM1A0576	PATLOLLA PRERANA REDDY										
18	18QM1A0577	PEDDA KURVA SAI										
19	18QM1A0578	PISKE AMULYA										
20	18QM1A0579	POLA VINAY KUMAR										
21	18QM1A0580	PULLANAGARI ARUN KUMAR REDDY										
22	18QM1A0581	PUSTHAKALA DIVYA										
	18QM1A0582	PYARASANI										

23		JYOTSNA										
24	18QM1A0583	R MOUNIKA										
25	18QM1A0584	R SAIRAM REDDY										
26	18QM1A0585	R SOUNDARYA										
27	18QM1A0586	RAM PRAKASH NAIK										
28	18QM1A0587	RAMAVATH VAMSHI										
29	18QM1A0588	RAYYAN AHMED										
30	18QM1A0589	ROTIKADI SWATHI										
31	18QM1A0590	S GEETHA SARANYA										
32	18QM1A0592	SANNAREDDY HEMANTH										
33	18QM1A0593	SARA HANISH										
34	18QM1A0594	SILVARI SAIRAM										
35	18QM1A0595	SINGANAMONI MOHAN										
36	18QM1A0596	SOMA ANUSHA YADAV										
37	18QM1A0597	SRI RAMULA SAI CHANDRA										
38	18QM1A0598	SUNIGANTI RAJESHREDDY										
39	18QM1A0599	SURAKANTI RUCHITA REDDY										
40	18QM1A05A0	SURAMPUDI GEETIKA										
41	18QM1A05A1	SUSHANTH BABU TAMMALI										
42	18QM1A05A3	T MANI TEJA										
43	18QM1A05A4	TADAKALA TANUJ										
44	18QM1A05A5	TATIPAMULA UTTAM KUMAR										
45	18QM1A05A6	THAKUR ABHISHEK SINGH										
46	18QM1A05A7	THANNIRU ANKAMRAJU										
47	18QM1A05A8	THUMMA MAHITHA										
48	18QM1A05B1	UPPALA NIKHITHA										
49	18QM1A05B2	VADARI DHRUVA										
50	18QM1A05B4	VINEELA SIDDINENI										
51	18QM1A05B7	YAKARA ASHWINI										

52	18QM1A05B8	YALALA HARISH KUMAR										
53	18QM1A05B9	YEMULA SAI SATYA										
54	18QM1A05C0	GOLLAGUDEM SHRAVANTHI										
55	19QM5A0501	CHINTHALA APARNA										
56	19QM5A0502	CHINTHALA VENUGOPAL SAGAR										
57	19QM5A0503	DEVARAKONDA SANATH KUMAR										
58	19QM5A0504	GANGAM SANJANA										
59	19QM5A0505	KONTHAM PRASHANTH REDDY										
60	19QM5A0506	KUMBER SAIKUMAR										
61	19QM5A0507	MADEPELLI SADHANA										
62	19QM5A0508	MD SHOAIB										
63	19QM5A0509	MOHAMMED AKHILUDDIN										
64	19QM5A0510	RONTALA SAI KUMAR										
65	19QM5A0511	SHAMSHABAD AKSHITHA										
66	19QM5A0512	SUDDYREDDY VAMSHIDHAR REDDY										

HOD

KG REDDY College of Engineering & Technology

KG Reddy College of Engineering & Technology
(Approved by AICTE, New Delhi, Affiliated to JNTUH, Hyderabad)
Chilkur (Village), Moinabad (Mandal), R. R Dist, TS-501504

Accredited by NAAC

26/30

CERTIFICATE COURSE EXAM
II B.TECH I SEM CSE
ADVANCED C PROGRAMMING

NAME M. Akhila HALL TICKET NO 1 8 Q M 1 A 0 5 6 4

Answer all the questions. All questions carry equal marks. Time: 1 Hrs. 30 marks.
I choose correct alternative:

1.	A C variable name can start with a ____			[C]
A. Number	B. Plus Sign (+)	C. Underscore	D. Asterisk (*)	
2.	Which of the following is the correct order of evaluation for the below expression? z = x + y * z / 4 % 2 -1			[D]
A. * / % + - =	B. = * / % + -	C. / * % - + =	D. * % / - + =	
3.	Which of the following are unary operators in C? 1 .! 2. Size of 3 .~ 4.&&			[D]
A. 1,2	B. 1,3	C. 2,4	D. 1,2,3	
4.	In which order do the following gets evaluated 1. Relational 2.Arthematic 3. Logical 4.Assignment			[A]
A. 2134	B. 1234	C. 4321	D. 3214	
5.	Name the loop that executes at least once.			[C]
A. For	B. While	C. Do-while	D. if	
6.	Who is known as the father of C Language?			[C]
A James A. Sosling	B. Vjarne Stroustrup	C. Dennis Ritchie	D. Dr. E. F. Codd	
7.	What is the work of **break** keyword?			[C]
A. Halt execution of program	B. Restart execution of program	C. Exit from loop or switch statement	D. None of the above	
8.	Which operators are known as Ternary Operator?			[B]
A. ::, ?	B. ?, :	C. ?, ;;	D. None of the above	
9.	An array elements are always stored in _____ memory locations			[A]
A. Sequential	B. Random	C. Sequential and Random	D. None of the above	
10.	What is right way to Initialization array?			[A]
A. int num[6] = { 2, 4, 12, 5, 45, 5 } ;	B. int n{} = { 2, 4, 12, 5, 45, 5 } ;	C. int n{6} = { 2, 4, 12 } ;	D. int n(6) = { 2, 4, 12, 5, 45, 5 } ;	

11.	What is (void*) ?			[B]
A. Representation of NULL pointer	B. Representation of void pointer	C. Error	D. None of above	
12.	Can you combine the following two statements into one? Char *p; P=(char*) malloc(100);			[D]
A. char p = *malloc(100);	B. char p = *malloc(100);	C. char *p = (char*)malloc(100);	D. char *p = (char *)(malloc*)(100);	
13.	What are Nibble, Word and Byte in computer language.?			[C]
A. Byte = 8 bits, Word= 4 Bytes, Nibble= 8 Bytes	B. Byte = 8 bits, Word=2 Bytes, Nibble=4 Bytes	C. Byte = 8 bits, Word=12 bits, Nibble=32 Bits	D. Byte = 8 bits, Word=24 bits, Nibble=40 Bits	
14.	What is the operator used to make 1's One's compliment.			[C]
A. & Bitwise AND Operator	B. | Bitwise OR operator	C. ~ Bitwise Negate Operator	D. ^ Bitwise Exclusive OR	
15.	What is the output of Bitwise OR operation | on (0110 | 1100).?			[A]
A. 1110	B. 1100	C.1000	D. 1010	
16.	Which of the following are C preprocessors?			[D]
A #ifdef	B. #define	C. #endif	D. all of the mentioned	
17.	#include statement must be written			[A]
A. Before main()	B. Before any scanf/printf	C. After main()	D. It can be written anywhere	
18.	The C-preprocessors are specified with which symbol?			[A]
A. #	B. $	C. " "	D. &	
19.	What is #include directive?			[A]
A. Tells the preprocessor to grab the text of a file and place it directly into the current file	B. Statements are not typically placed at the top of a program	C. All of the mentioned	D. None of the mentioned	
20.	The preprocessor provides the ability for what /			[D]
A. The inclusion of header files	B. The inclusion of macro expansions	C. Conditional compilation and line control	D. All of the mentioned	

II Fill in the Blanks:

21.	C programs are converted into machine language with the help of compiler
22.	An Array is a collection of data elements of the same data type
23.	A structure is a group of data elements that may have different data types
24.	A Preprocessor directive begin with a # symbol
25.	main() { int a; a = 30; a = a << 2 ; printf("a = %d",a);}Answer: 32
26.	main() { int x , y; x = 2003; x++; y = x++; y = x; y++; x--; printf ("%d%d",x,y);Answer: 2004, 2006
27.	%.C Format Specifier can be used to read/print the character
28.	Repeating a set of statements for a specific number of times is called looping structure.
29.	C program execution begins from main function
30.	a=10; b=++a; then the value of b is 11