Module-2 theory

Ch-2 introduction programming

Q-1 history of c

Ans: C language create in 1972 by richard dennis. This is pop language. today use c language because this language pillor of all

Programming language. Today use c language in small level operating system and create many game by c.

Q-1(LAB Exercise)

C language use in embedded system,operating system,and,game

There are 3 example for use of c.

1.Automative control system:-

Engine control units (ECUs), anti-lock braking systems (ABS), airbag controllers, infotainment systems.

2.opreating system:-

Building kernel modules, drivers, file systems, memory management, and system utilities.

Ex-unix/linx system

3.game devlopment:-

C language use in game management for portability,low access,fast and efficient.

Ex-Doom,Quake,Wolfenstin 3D etc..

Q-2 Setting up environment

Theory exercise ans:

Download vscode by chrome. In window category. After then open vscode. And download Five consol.

1.c/c++

2.c/c++ comipler run

3.c/c++ extension pack

4.c/c++ themes

5.code runner

Devc++

Devc++ installing from chrome. Fun with mingw compiler.

This is outdated and use many compiler for programming.

Q-3

Basic structure of c

**->** including headers

There are two type of header file.

1.#include<stdio.h>

Standard input header file.

2.#include<conio.h>

Console input header file.

**->** Main function

It’s entry point for every program in programming language.

Ex- C language Main function

#include<stdio.h>

Int main()

{

return 0;

}

**->** data type :-

It’s used all type value of storge in one function it’s called data type.

There are 4 type of data type:

1.integer : store numeric value. Ex :- 1,3,4,8,etc…

2.float : store decimal value. Ex :-

3.char

4.double

**->** Variable :-

A variable is a named memory location used to store data that can change during program execution.

Example:-

Int age=20;

Float salary=5000.50;

Char grade = ‘A’;

Start with letter or underscore.

Can contain letters,digits,underscore.

Case sensitive and cannot use keywords.

Comments

Comments are notes written in a program to explain the code. They are ignored by the compiler.

1.Single line comments

#include <stdio.h>

int main()

{

char name[50]; // single-line comment

printf("Enter name: ");

scanf("%s", name);

printf("Name: %s", name);

return 0;

}

2. Multi line comments

#include <stdio.h>

int main()

{

/\* This is a

multi-line comment

explaining the program \*/

char name[50];

printf("Enter name: ");

scanf("%s", name);

printf("Name: %s", name);

return 0;

}

Q-4 Opreator

An specific symbol for assign special value and use for specific operation.

1.airthmetic operator

Used for mathematical formula

+,-,\*,%,/ etc…

2.relational operator

Used for compared two component.

1. < - less than - a<b - a is less then b

2. <= - less than or equal to - a<=b - a is less then equal to b

3. > - greater than - a>b - a is greater then b

4. >= - greater than equal to - a>=b - a is greater then equal to b

5. == - equal to - a==b - a is equal to b

6. != - not equal to - a!=b - a is not equal to b

3. Logical operator

This condition used for combain two or more condition is true/false.

Logical And (&&)

Return true if both are true.

(a>5 && b<10)

Logical not(!)

It’s used for reverse the logical value of condition.

Ex:-

#include<stdio.h>

Int main()

{

Int n=5;

if(!(n>10))

{

printf("n is NOT greater than 10\n");

}

return 0;

}

Ex-5 is not greater than 10

4. Assignment operator

This operator used for assign value

EX:- +,+=,\*=,/=,%=

5. Bitwise opreator

&-bitwise and operator

|-bitwise or operator

^-bitwise executive or

<< - left shieft operator

>> - right sheift operator

6. Conditional operator

Conditional operator ? extanssion: expression;

(age>=18)?printf(“vote-yes”):printf(‘vote”);

7.special operator

& - address operator - it is used to determine variable. Address others.

\* - pointer operator - it’s used to declare get value from it.

, - comma operator - it’s used for declare get value from pointer variable.

Size of - operator

Q-5 control flow statement

There are many type of control flow statement in programming language.

1. If else

* it ‘s check for condition true/false.

Ex -

#include<stdio.h>

Int main()

{

int age;

printf(“Enter age: “);

scanf(“%d”,&age);

if(age>=18)

{

printf(“eligible for vote”);

}

else

{

printf(“not to eligible to vote”)

}

return 0;

}

* output

Enter age: 18

Eligible for vote

2. else if

* It’s used for check multiple condition.

Ex-

#include<stdio.h>

Int main()

{

int num;

printf(“Enter num: “);

scanf(“%d”,&num);

if(num>0)

{

printf(“positive number”);

}

else if(num<0)

{

printf(“Negative number”);

}

else

{

printf(“Zero”);

}

return 0;

}

3. nested if

This condition used for multi level condition for decision making and get result by another condition.

Ex-

#include<stdio.h>

int main()

{

int a,b,c;

printf(“Enter a: “);

scanf(“%d”,&a);

printf(“Enter b: “);

scanf(“%d”,&b);

printf(“Enter c: “);

scanf(“%d”,&c);

if(a>b)

{

if(a>c)

{

printf(“A is max”);

}

}

else

{

if(b>c)

{

printf(“B is max”);

}

else

{

printf(“C is max”);

}

}

return 0;

}

* output

Enter a: 1

Enter b: 3

Enter c: -4

B is max

Q-5 looping

It’s one type of function one condition run many time it’s called loop.

There are two type of loop

1. Entry controlled loop - for,while

2.Exit controlled loop - do-while

1.Entry controlled loop

This loop run before execution.

1.for(initialization;condition;incre/decre)

2. while(condition)

{

incre/decre;

}

1. Exit controlled loop

This loop run after execution.

do

{

incre/decre;

}

while

At least one time run this loop.

Q-6 loop control statement

1. break

This statement used for stop loop.

Ex-

#include<stdio.h>

int main()

{

int i;

for(i=1;i<=10;i++)

{

if(i==3)

{

break;

}

printf("%d ", i);

}

}

* output

1 2

1. Continue

This statement run all loop number set specific number.

Ex-

#include<stdio.h>

int main()

{

int i;

for(i=1;i<=10;i++)

{

if(i==3)

{

continue;

}

printf("%d ", i);

}

}

* output

1 2 4 5 6 7 8 9 10

1. goto statment

This statement used for jump one statement to another statement.

Ex-

#include<stdio.h>

int main()

{

int i=1;

loop:

if(i<=5)

{

printf(“%d”,i);

i++;

goto loop;

}

return 0;

}

* output

1 2 3 4 5

Q-8 function in c

Function is a set of code which only runs when it is called function.

1. Library function

It have already defined in library.

Ex- printf()-used for print

scanf()-used for store value.

sqrt()-used for squareroot

1. Userdefined function

It is create by user for run specific task.

Parts of userdefined function

1.Function declaration(prototype)

2.Function definition

3.Function call

There are 4 part of user defined function

1. Without parameter without return type

#include<stdio.h>

void say()

{

printf(“Hello”);

}

Int main()

{

say();

return 0;

}

* Output

Hello

2. With parameter with return type

#include <stdio.h>

int add(int a,int b)

{

return a+b;

}

int main()

{

int x,y,ans;

printf("Enter x: ");

scanf("%d",&x);

printf("Enter y: ");

scanf("%d",&y);

ans=add(x,y);

printf("\n %d + %d = %d",x,y,ans);

return 0;

}

3.without parameter with return type

Ex-

#include<stdio.h>

int getvalue()

{

return 100;

}

int main()

{

int value;

printf(“Enter value: ”);

scanf(“%d”,&value);

value=getvalue();

printf(“value is : %d”,value);

return 0;

}

* Output

Enter value: 67

value is: 100

4. Without return type with parameter

Ex-

#include<stdio.h>

void sum(int a,int b)

{

int sum=a+b;

printf(“sum : %d”,sum);

}

int main()

{

int r,t;

printf(“Enter r: “);

scanf(“%d”,&r);

printf(“Enter t: “);

scanf(“%d”,&t);

sum(r,t);

return 0;

}

* Output

Enter r: 56

Enter t: 112

Sum: 168

Q-9 Array

Collection of elements of the same data type store in same memory location.

Two type of array initialization

1.compile time initialization

Assign value at the time of declaration before the program it’s called compile time initialization.

Ex - int arr[5] = {5,6,7,8,9,10}

2. Run time initialization

We save all value store during program it’s called run time initialization.

All value give by user or calculate the value during program.

There are 2 type of array

1.one dimonsianal array

Ex- int arr[5];

2. Two dimonsianal array

Ex- int a[3][3],b[3][3];

Q-9 Pointer

A variable that can store memory address it’s called pointer.

int num=90;

int\*ptr;//pointer declareation

ptr=&num;// ptr store number address

Ex -

#include<stdio.h>

int main()

{

int num =90;

int\*ptr;

ptr=&num;

printf(“%d”,num);

return 0;

}

Q-11 String

This is collection of character array it’s called string.

Ex-

#include<stdio.h>

int main()

{

char arr[10]="Romit";

printf("Name: %s",arr);

return 0;

}

* Strlen

It’s used for calculate string length.

Ex-

#include <stdio.h>

#include <conio.h>

int main()

{

char str[20];

int a;

printf("Enter string: ");

gets(str);

a=strlen(str);

printf("Length of string: %d",strlen(str));

return 0;

}

-> Output

Enter string: Romit

Length of string: 5

* strcpy

This string used for copy any string.

Ex-

#include <stdio.h>

#include <conio.h>

int main()

{

char str1[20];

char str2[20];

printf("Enter string: ");

gets(str1);

strcpy(str2,str1);

printf("Copied string : %s",str2);

return 0;

}

-> Output

Enter string:Romit

Copied string:Romit

* strcat

This string contained two string it’s called strcat

Ex-

#include <stdio.h>

#include <conio.h>

int main()

{

char str1[20];

char str2[20];

printf("Enter string: ");

gets(str1);

printf("Enter string: ");

gets(str2);

printf("Containted string : %s",strcat(str1,str2));

return 0;

}

-> Output

Enter string:hello

Enter string:World

Containted string:helloWorld

* Strcmp

This function compare two string it’s called strcmp.

Ex-

#include <stdio.h>

#include <conio.h>

int main()

{

char str1[20];

char str2[20];

int b;

printf("Enter string: ");

gets(str1);

printf("Enter string: ");

gets(str2);

b=strcmp(str1,str2);

if(b==0)

{

printf("\n String are same");

}

else

{

printf("\n String are not same");

}

return 0;

}

->output

Enter string:romit

Enter string:romit

String are same

* Strchr

This function is used for findout specific character it’s called

Strchr.

Ex-

#include <stdio.h>

#include <string.h>

int main()

{

char str[30];

char ch;

char \*ptr;

printf("Enter a string: ");

gets(str);

printf("Enter a character to find: ");

scanf("%c", &ch);

ptr = strchr(str, ch);

if (ptr != NULL)

printf("Character '%c' position: %d", ch, ptr - str + 1);

else

printf("Character '%c' not found.", ch);

return 0;

}

->output

Enter a string:romit

Enter a character to find:r

Character ‘r’ position : 1

Q-12 Structure

This is a user defined function it’s store different type of data store in one shat it’s called structure.

It’s useful for one type of data store in one shet.

Ex-

#include<stdio.h>

#include<conio.h>

struct student

{

int id;

char name[190];

}s1;

int main()

{

printf("\n Enter Student Details");

printf("\n Enter Id: ");

scanf("%d",&s1.Id);

printf("\n Enter name:");

scanf("%s",&s1.name);

printf("\n Your Id : %d , Your name : %s",s1.Id,s1.name);

return 0;

}

-> Output

Enter Id:105

Enter name:romit

Your Id:105,Your name:romit

* Structure Declaration

struct student

{

int id;

char grade[10];

}s1;

* Initialization

Structure student s1 = {101,”A”};

Q-13 File handling

File handling is process of create file.

We create,open,read,file,write and close operation in c.

#include<stdio.h>

#include<conio.h>

int main()

{

FILE\* fptr;

fptr = fopen("file.txt", "w");

if (fptr == NULL)

printf("The file is not opened.");

else

printf("The file is created Successfully.");

return 0;

}

* Out put

Create file-file.txt

fprintf

It’s similar as printf. Standard function used for writing formatted output to a specified file stream.

fputs

Prints whole line in the file and new line at the end.

fputc

Print a single character into the file.

fopen

It’s used in opening file.

Tab 2

**Ch-1 Overview of It industry**

**Q-1 Program**

**The set of Instruction.**

**Lab Exercise**

**C python**

**#include<stdio.h>**

**Int main() print(“Hello world”)**

**{**

**printf(“Hello World”);**

**return 0;**

**}**

**Theory Exercise**

**Hello word program run by function like this**

**Stdio is headerfile all c language start by header file program**

**Int main is user defined function because program start by function**

**Printf is library function because it is provide print output.**

**Hello world program run easily in python because print is inbuilt function in python language.**

**Q-2**

**Programming**

**This is process of giving a computer step by step code and as output solve problem and get output**

**1.Understand problem**

**2.create algorithm for given problem**

**3.algorithem convert into code(in c,python,java etc)**

**4.compile/interpreter-: c,c++,java code compile**

**Python language code interpreted.**

**5.testing:-check code for error**

**6.result:- show output**

**Q-3**

**Types of Programming Language**

**High language Low language**

**Easy to read and write hard to read closer to**

**For programmer. Machine language.**

**Need less knowledge Assembly language and**

**For hardware. Machine code.**

**Highly portable Hardware architecture is hard**

**Across different system. For understanding.**

**Compiler and interpreter convert Low portability; tied to**

**Into machine code. Specific hardware.**

**Generally slower due to Executed directly by the cpu**

**Or through assembler.**

**interpration/translation.**

**Use automatic system for Faster execution closer to**

**Memory and management. Cpu instructions.**

**English and easy for programmer. Manual programmer**

**manages.**

**App development,web development Cryptic,Symbolic,and**

**Software development etc. numeric instruction.**

**Ex- C,java,python etc. System programming,**

**Embedded system,device**

**Driver.**

**Role about client and server in web communication**

**Client role**

**1.Send request(http,https,ftp)**

**2.Break request into segments(TCP/UDP).**

**3.Adds IP address(Source&destination)**

**4.Converts IP packets->frams(MAC address)**

**5.Sends raws bites overcable/wireless.**

**Server role**

**1.Recives request,process it,proper response.**

**2.Reassembles segments into complete message**

**3.user IP to know where to send response.**

**4.Recevice frams and extracts packets.**

**5.Receives raw bets and converts back**

**TCP/IP model layers**

**There are 4 layers in TCP/IP model**

**Application layer**

**Provides network service directly user for run application.**

**And interact with network.**

**EX-http,https,smtp,ftp.**

**Transport layer**

**The transport layer ensures end to end communication between application on different device.**

**Ex-TCP,UDP**

**Internet layer**

**The internet layer is responsible for logical addressing and ruting.**

**How to get the across different network.**

**Ex-IP address,router,packet.**

**Network Access layer**

**Network access layer is lowest layer of tcp/ip model.**

**It is physical transmission of data move from one device to each other device.**

**Ex-wifi/ethrent physically device frames.**

Tab 3

**Role about client and server in web communication**