**UNIONS:**

**Only one member can be accessible at a time.**

**EX:** union item

{ int m;

float x;

char c;

};

Size of the union will be the datatype range that has higher range compare to another datatypes in a union.

**EX:**

#include <stdio.h>

#include<string.h>

#include<stdlib.h>

typedef union sEmployee

{

int id;

float sal;

char g;

char Name[20];

}sEMP;

typedef union uEmployee

{

int id;

int sal;

char g;

char Name[20];

}uEMP;

int main() {

sEMP es;

uEMP eu;

uEMP \*eu1=NULL;

printf("\nSize of Union = %d",sizeof(es));

printf("\nSize of Union = %d",sizeof(eu));

eu.id=101;

eu.sal=10000;

strcpy(eu.Name,"Sreeja");

printf("\nName = %s",eu.Name);

eu1=(uEMP \*)malloc(sizeof(uEMP));

eu1->id=102;

printf("\nID: %d",eu1->id);

printf("\n\n");

return 0;

}

**ENUM: (Enumeration)**

The keyword enum signals the start of an enumeration type.

**Syntax:** enum enum-type-name{enumeration list}variable\_list;

Enum is starting from starting index(0).

**EX:**

#include <stdio.h>

enum RegClrCodes

{

Black,

Brown,

Red,

Orange,

Yellow,

Green,

Blue,

Violet,

Gray,

White

}RCC;

int dispMenu()

{

int cc;

printf("\nPress,");

printf("\n0. Black");

printf("\n1. Brown");

printf("\n2. Red");

printf("\n3. Orange");

printf("\n4. Yellow");

printf("\n5. Green");

printf("\n6. Blue");

printf("\n7. Violet");

printf("\n8. Gray");

printf("\n9. White");

printf("\nEnter the colorcode:");

scanf("%d",&cc);

return cc;

}

int main() {

switch(dispMenu())

{

case Black:

printf("\nBlack");

break;

case Brown:

printf("\nBrown");

break;

case Red:

printf("\nRed");

break;

case Orange:

printf("\nOrange");

break;

case Yellow:

printf("\nYellow");

break;

case Green:

printf("\nGreen");

break;

case Blue:

printf("\nBlue");

break;

case Violet:

printf("\nViolet");

break;

case Gray:

printf("\nGray");

break;

case White:

printf("\nWhite");

break;

default:

printf("\nEnter the valid Color\n");

}

printf("\n\n");

return 0;

}

**EX:**

#include <stdio.h>

enum RegClrCodes

{

Black=101,

Brown,

Red,

Orange,

Yellow=303,

Green,

Blue,

Violet,

Gray,

White

}RCC;

int dispMenu()

{

int cc;

printf("\nPress,");

printf("\n%d. Black",Black);

printf("\n%d. Brown",Brown);

printf("\n%d. Red",Red);

printf("\n%d. Orange",Orange);

printf("\n%d. Yellow",Yellow);

printf("\n%d. Green",Green);

printf("\n%d. Blue",Blue);

printf("\n%d. Violet",Violet);

printf("\n%d. Gray",Gray);

printf("\n%d. White",White);

printf("\nEnter the colorcode:");

scanf("%d",&cc);

return cc;

}

int main() {

switch(dispMenu())

{

case Black:

printf("\nBlack");

break;

case Brown:

printf("\nBrown");

break;

case Red:

printf("\nRed");

break;

case Orange:

printf("\nOrange");

break;

case Yellow:

printf("\nYellow");

break;

case Green:

printf("\nGreen");

break;

case Blue:

printf("\nBlue");

break;

case Violet:

printf("\nViolet");

break;

case Gray:

printf("\nGray");

break;

case White:

printf("\nWhite");

break;

default:

printf("\nEnter the valid Color\n");

}

printf("\n\n");

return 0;

}

**FILE HANDLING:**

A collection of logically related information.

**Two Types:**

**1.Seqential File:** All records are arranged in a particular order.

**2.Random Access File:** Files are arranged at random

**Declaration:** FILE \*fp

FILE \*fopen(char \*name,char \*mode);------if the file is present open the file

fp=fopen(“file name”,”mode”);-----mode of the file---read only, write only etc.

Once if the file is able to open it will give the address of the particular file.

**EX:**

FILE => DS

Fd => file pointer to FILE DS

Fopen(“NameofFILE”, “Mode”) => opening a file

Fclose(fd) => close the opened file

Read/write

fprintf,fscanf => formatted writing and reading resp.

fputs,fgets => formatted i/o operation

fwrite/fread => reading/writing binary objects.

#include <stdio.h>

#include<string.h>

#include<stdlib.h>

int main() {

FILE \*fd=NULL;

fd = fopen("Test1.txt","w");

if(fd == NULL)

{

perror("fopen: ");

printf("\n%p\n\n",fd);

exit(1);

}

printf("\nOpened the file successfully\n\n");

fclose(fd);

return 0;

}

**EX:**

#include <stdio.h>

#include<string.h>

#include<stdlib.h>

int main() {

FILE \*fd=NULL;

char ch='a';

char Name[]="Sreeja";

char rName[100];

fd = fopen("Test1.txt","w");

if(fd == NULL)

{

perror("fopen: ");

printf("\n%p\n\n",fd);

exit(1);

}

printf("\nOpened the file successfully\n\n");

/\* while((ch=fgetc(fd))!=EOF)

{

putchar(ch);

}

\*/

/\* fputc(ch,fd);

fputc(ch,fd);

fputc(ch,fd);

fputc(ch,fd);

fputc(ch,fd);\*/

/\*

for(int i=0;i<strlen(Name);i++)

fputc(Name[i],fd);

\*/

i=0;

while((ch=fgetc(fd))!=EOF)

{

rNmame[i]=ch;

i++;

}

rName[i] = '\0';

printf("\nrName = %s\n",rName);

return 0;

}

COPYING THE FILE:

#include <stdio.h>

#include<stdlib.h>

int main() {

FILE \*rfptr=NULL;

FILE \*wfptr=NULL;

char ch;

rfptr=fopen("Test13.txt","r");

if(rfptr == NULL)

{

perror("fopen: ");

exit(1);

}

wfptr=fopen("Test1.txt","w");

if(wfptr == NULL)

{

perror("fopen: ");

exit(1);

}

while(ch=fgetc(rfptr) != EOP)

{

fputc(ch,wfptr);

}

fclose(wfptr);

fclose(rfptr);

return 0;

}

**Read the contents from the database the record of the employes.**

101|Amit Kumar|M|88888|1000000

101|Amit Kumar|M|88888|1000000

101|Amit Kumar|M|88888|1000000

My using system we cannot execute linux commands.