/\* Storage classes

A variable has four properties

1. Memory location

2. Initial Value

3. Scope of the variable

4. Lifetime of the variable

On the basis of above four properties we have four storage classes

1. Automatic storage class

Reserved Word - auto

1. Main memory

2. Garbage

3. Local to the block declared in

4. Till the control remains in that block

eg:

main()

{

auto int a;/int a;

}

2. Register Storage class

Reserved Word - register

1. CPU Register

2. Garbage

3. Local to the block declared in

4. Till the control remains in that block

main()

{

register int a;

}

3. Extern Storage class

Reserved Word - extern

1. Main memory

2. Garbage

3. Throughout the program

4. Till the program execution ends

extern int a;/int a;

main()

{

}

4. Static Storage class

Reserved Word static

1. main memory

2. zero

3. Local to the block declared

4. Persistent between different function calls

\*/

void show();

main()

{

show();

show();

show();

}

void show()

{

int x=6;

printf("%d\n",x);

x++;

}

/\* Pre Processor

#include<> to include files

#define to define constants and macros

#undefine to undefine them

#ifdef to check whether defined or not

#else else of if

#endif to end the if

#define city karnal

#define max 5

main()

{

int a[max];

for(index=0;index<max;index++)

}

\*/

#define max 5

#define area(x) 3.14\*x\*x //MACRO DEFINITION

main()

{

printf("%f\n",area(3));

printf("%f\n",area(3.5));

#ifdef max

printf("its there");

#else

printf("its not there");

#endif

#undef max

#ifdef max

printf("its there");

#else

printf("its not there");

#endif

}

//Difference between function and macro

//These occupy memory space only once before the processing start

//These are data type independent

//These do not support control structures

/\*

difference

if and switch

while and do while

array and structure

CBV CBR

macro and function

structure union and enumeration

\*/

/\* enumeration is a user defined data type. It is used to create a set of values

0. enum boolean {true, false};

1. enum vowels{a,e,i,o,u};

2. vowels x;

3. x=a;

4. if(x==e)

5. for(x=a;x<=u;x++)

\*/

enum designation {peon, clerk, manager, director};

struct Employee

{

int empno;

char name[20];

enum designation d;

};

main()

{

struct Employee e;

int ch;

//make a choice for designation

printf("Enter empno");

scanf("%d",&e.empno);

fflush(stdin);

printf("Enter name");

gets(e.name);

printf("Enter your choice \n0. peon \n1. clerk \n2.manager \n3.director");

scanf("%d",&ch);

if(ch==0)

e.d=peon;

else

if(ch==1)

e.d=clerk;

else

}

//complete structure program

//https://www.geeksforgeeks.org/enumeration-enum-c/