C Programming Concepts

Kaushal Kishor Sharma
Advanced Computing Training School (ACTS)
CDAC, Pune

typedef:

```
typedef int INT;
typedef unsigned int UI;
typedef int * IP;  IP ptr;

typedef struct Student
{    char name[20];
     int  rollNo;
     float  accBal;
}record;
record stud1, stud2;
record *ptr;
```

```
typedef int INT;
unsigned INT a;    // wrong

typedef struct student sType;
struct sType;          // wrong

typedef struct {
        tnt member1;
        float member2;
}newType;
newType var1, var2;
```

```
int fadd(int, int);
typedef int (*ptype)( int, int);
ptype pf1, pf2;
pf1=fadd;   pf1(10,20);

typedef int (*arrayPtr) [5];
arrayPtr  aptr1;

typedef void (*fptr) (int a, void (*) (float *));
fptr fp1;
```

Preprocessor: #define

```
#define MAX 100
#define SIZE  10
#define AND &&
#define BEGIN int main(void) {
#define END }
#define NEWLINE  printf("\n");
#define INFINITE  while(1);
#define MSG "hello world";
```

Preprocessor:
#parameterized Macros

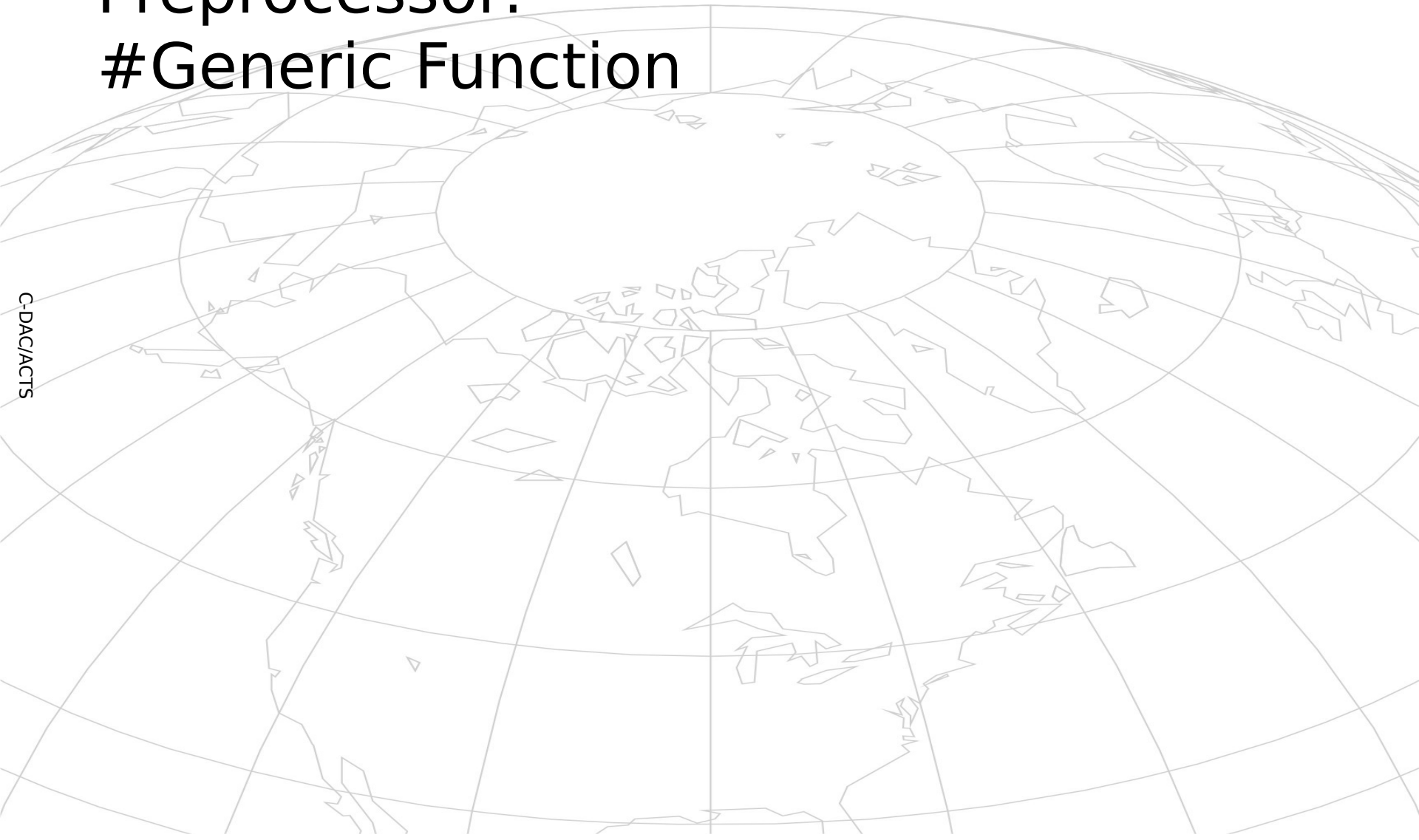
```
#define MUL(a,b)  ((a) * (b))
#define MINVAL(a,b) ((a) < (b) ? (a) : (b))
#define ISLOWER(ch) (ch>=97 && ch<=122)

#define MUL(a,b) a*b          // it can create prob
#define SQRT(a)   a*a         // it can create prob

MUL(1+2, 2+3);
SQET(a++);
```

Preprocessor: #Nested Macros

```
#define ISLOWER(ch) (ch>=97 && ch <= 122)
#define ISUPPER(ch) (ch >= 65 && ch <=90)
#define ISALPHA(ch)  ISLOWER(ch) || ISUPPER(ch)
```

```
#define SWAP(type, a, b) {type t ; t=a; a=b ; b=t ;}
Int main()
{
    int x=10, y=20;
    SWAP(int, x,y);
    printf("x=%d y=%d", x,y);
}
```

Preprocessor:
#Generic Function



5

```
#define SQRT(FNAME, DTYPE)  \
        DTYPE FNAME(DTYPE X) \
        {                    \
                return X*X;  \
        }
SQRT(sqrt_int, int)
SQRT(sqrt_float, float)
SQRT(sqrt_double, double)
int main()
{       printf("%d\n", sqrt_int(10));
        printf("%f\n", sqrt_float(10.1));
        printf("%lf\n", sqrt_double(10.12));


}
```

Preprocessor:

-Token Pasting Operator(##)

-Stringizing Operator(#)

6

```
#define CONCAT(a,b) a##b
#define PRINT(var,format)  printf(#var"=%"#format"\n", var);

int main()
{
  int ab=20;
  char str[]="abcde";
  printf("a##b = %d\n",CONCAT(a,b));
  (CONCAT(print,f))("vlaue of ab is %d\n", ab);
  PRINT(ab,d);
  PRINT(str,s);
}
```

Preprocessor:
-Preprocessor Symbols

```
int main()
{
  printf("File name: %s\n", __FILE__);
  #line 1000
  printf("On line no.: %d\n", __LINE__);
  printf("Inside function: %s\n", __FUNCTION__);
  printf("Compiled at: %s\n", __TIMESTAMP__);
  printf("Compiled on: %s\n", __DATE__);
  printf("Executed at: %s\n", __TIME__);
}
```

Preprocessor:

-Conditional compilation

#if

#ifdef - if defined

#ifndef - if not defined

#elif - else if

#else

#undef

8

```
#define ARCH_32
#define FLAG  1
int main()
{       #ifdef ARCH_32
          printf("this is for 32 bit\n");
        #else
          printf("this is for 16 bit\n");
        #endif
        #if FLAG==1
          printf("FLAG is true\n");
        #else
          printf("FLAG is flase\n");
        #endif
}
```

Preprocessor directive :
- avoid multiple time inclusion of a single header file

#ifndef
#endif

#pragma once

Some time after applying this all to our header files, still same problem comes, then in case compile header files (eg gcc -c test.h, gcc -c a.h, gcc -c b.h)