Structures

* To store different data types together
* Typedef struct \_\_\_\_\_\_\_\_\_ { } <alias> ;
* user defined data type

Union

* set 1 value for 1 data type
* shared memory for multiple parameters
* max memory length of the members s

Enum

* Set numbers to some key words
* Give a name to an integer
* enum <name> {a,b,c,d,e};
* here a = 0 b=1 c=2 ….

Functions

* top-down structured programming
* function prototyping
  + type <function name> (<arguments>) ;
* formal parameter
  + the one in function header file
* actual parameters
  + the value that is being passed
* int -> call by value
* pointers -> call by reference
* reference variables
  + you pass the address of the actual variable
  + function can return reference
  + int & <function name> (\_.....
* Inline function
  + Reduce the time cost
  + inline int <function name> (…) ………
  + replaces the function call by the code snippet
  + request to the compiler
* default arguments
  + int emp(int id, int room\_number = 103){}
  + can change while passing
  + extreme left while declaring function
* constant argument
  + avoid editing of passed value
  + usually for pointers