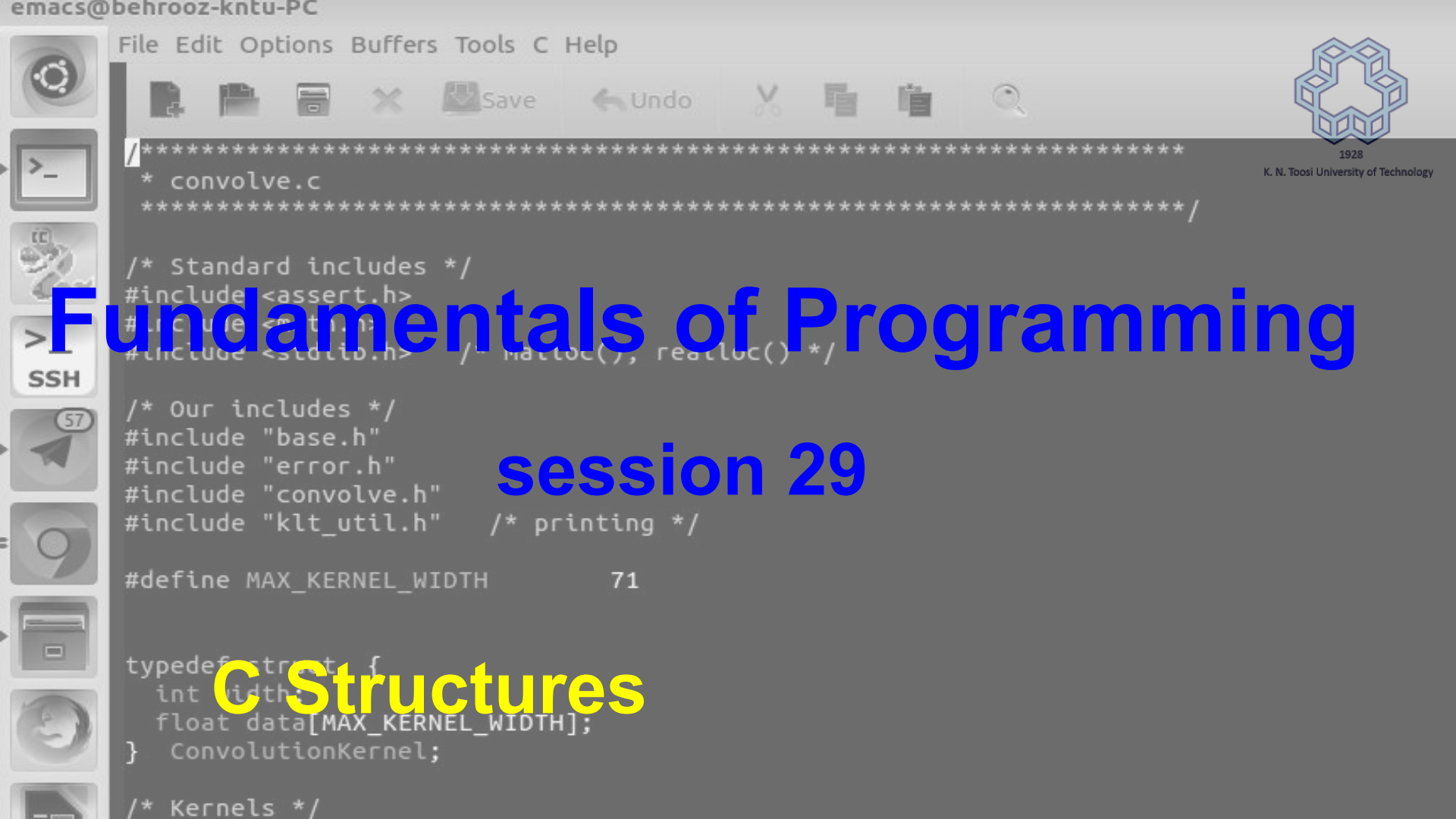
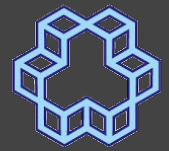
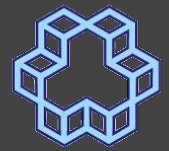

emacs@behrooz-kntu-PC
File Edit Options Buffers Tools C Help
Save
Undo
1928
K. N. Toosi University of Technology
/***
* convolve.c
***/
/* Standard includes */
#include <assert.h>
#include <math.h>
#include <stdlib.h> /* malloc(), realloc() */
/* Our includes */
#include "base.h"
#include "error.h"
#include "convolve.h"
#include "klt_util.h" /* printing */
#define MAX_KERNEL_WIDTH 71
typedef struct {
int width;
float data[MAX_KERNEL_WIDTH];
} ConvolutionKernel;
/* Kernels */
Fundamentals of Programming
session 29
C Structures

C Structures



- Defining new types
- A Student type
- Student has:
  - first name
  - last name

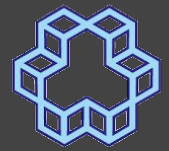
C Structures



- Defining new types
- A Student type
- Student has:
  - first name
  - last name

```c
int main() {
  char *student[2];

  student[0] = "Ali";
  student[1] = "Karimi";

  return 0;
}
```

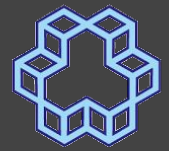
C Structures



- Defining new types
- A Student type
- Student has:
  - first name
  - last name
  - student id (?)
  - age (?)

```c
int main() {
  char *student[2];

  student[0] = "Ali";
  student[1] = "Karimi";

  return 0;
}
```

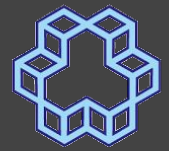
C Structures



- Defining new types
- A Student type
- Student has:
  - first name
  - last name
  - student id (?)
  - age (?)

```
struct Student {
    char firstName[20];
    char lastName[20];
    int id;
    int age;
    char gender;
};
```

C Structures



- Defining new types
- A Student type
- Student has:
  - first name
  - last name
  - student id (?)
  - age (?)

```
struct Student {
    char firstName[20];
    char lastName[20];
    int id;
    int age;
    char gender;
};
```

very important!

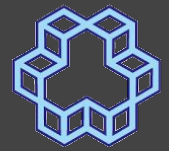
C Structures


```c
struct Student {
  char firstName[20];
  char lastName[20];
  int id;
  int age;
  char gender;
};
```

very important!

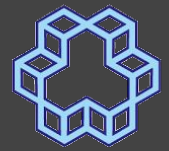
```c
int main() {
  struct Student ali, reza;

  ali.id  = 9612345;
  ali.age = 18;
  ali.gender = 'M';

  strcpy(ali.firstName, "Seyed Muhammad Ali");
  strcpy(ali.lastName, "Clay");

  return 0;
}
```

Passing structures to functions



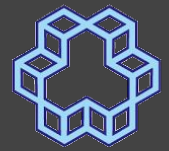

```c
void printStudent(struct Student s) {
  printf("First name: %s\n", s.firstName);
  printf("Last  name: %s\n", s.lastName);
  printf("ID:  %7d\n", s.id);
  printf("Age: %d\n", s.age);
  printf("Gender: %c\n\n", s.gender);
}
```

Passing structures to functions




```c
void printStudent(struct Student s);

int main() {
  struct Student ali, reza;

  ali.id  = 9612345;
  ali.age = 18;
  ali.gender = 'M';

  strcpy(ali.firstName, "Seyed Muhammad Ali");
  strcpy(ali.lastName, "Clay");

  printStudent(ali);

  return 0;
}
```

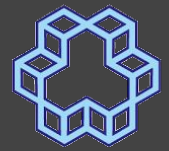
Passing structures to functions




```c
void printStudent(struct Student s);

int main() {
  struct Student ali, reza;

  ali.id  = 9612345;
  ali.age = 18;
  ali.gender = 'M';

  strcpy(ali.firstName, "Seyed Muhammad Ali");
  strcpy(ali.lastName, "Clay");

  printStudent(ali);

  return 0;
}
```

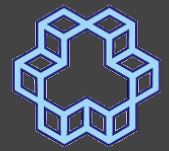
```
nasihatkon@kntu:code$ gcc struct2.c && ./a.out
First name: Seyed Muhammad Ali
Last  name: Clay
ID:  9612345
Age: 18
Gender: M
```

passed by value or by reference?




```
  printStudent(ali);
  printStudent(ali);

  return 0;
}


void printStudent(struct Student s) {
  printf("First name: %s\n", s.firstName);
  printf("Last  name: %s\n", s.lastName);
  printf("ID:  %7d\n", s.id);
  printf("Age: %d\n", s.age);
  printf("Gender: %c\n\n", s.gender);

  s.age++;
}
```

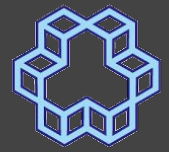
passed by value or by reference?




```
  printStudent(ali);
  printStudent(ali);

  return 0;
}


void printStudent(struct Student s) {
  printf("First name: %s\n", s.firstNa
  printf("Last  name: %s\n", s.lastNar
  printf("ID:  %7d\n", s.id);
  printf("Age: %d\n", s.age);
  printf("Gender: %c\n\n", s.gender);

  s.age++;
}
```

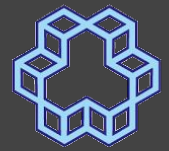
```
nasihatkon@kntu:code$ gcc struct4.c && ./a.out
First name: Seyed Muhammad Ali
Last  name: Clay
ID:  9612345
Age: 18
Gender: M

First name: Seyed Muhammad Ali
Last  name: Clay
ID:  9612345
Age: 18
Gender: M
```

structure as return value




```
struct Student ali, parvin;

ali    = createStudent("Ali", "Karimi", 9612345, 18, 'M');
parvin = createStudent("Parvin", "Etesami", 9612347, 111, 'F');

printStudent(ali);
printStudent(parvin);
```

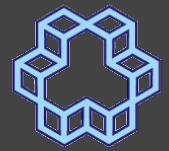
structure as return value




```
ali     = createStudent("Ali", "Karimi", 9612345, 18, 'M');
```

```
struct Student createStudent(char fName[], char lName[], int id, int age, char g) {
  struct Student s;

  strncpy(s.firstName, fName, 20);
  strncpy(s.lastName,  lName, 20);
  s.id  = id;
  s.age = age;
  s.gender = g;

  return s;
}
```

structure as return value

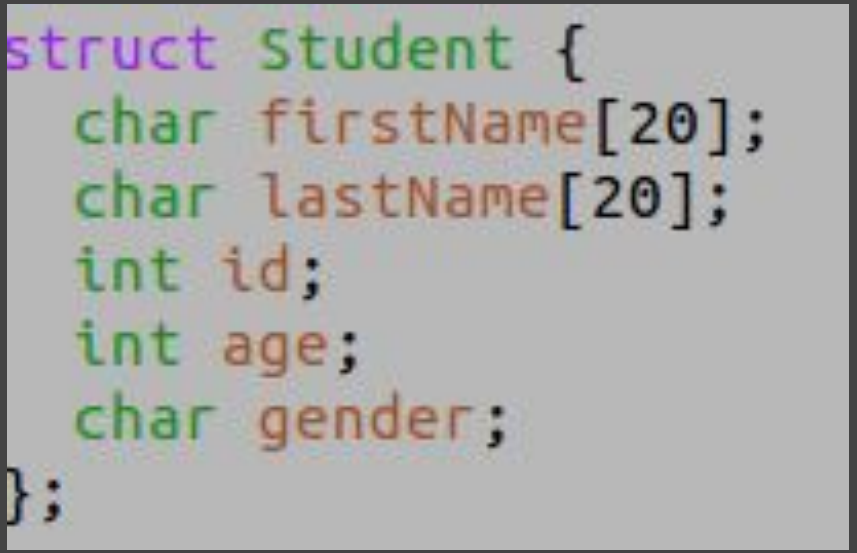



```
ali     = createStudent("Ali", "Karimi", 9612345, 18, 'M');
```

```
struct Student createStudent(
  struct Student s;

  strncpy(s.firstName, fName,
  strncpy(s.lastName,  lName,
  s.id  = id;
  s.age = age;
  s.gender = g;

  return s;
}
```

```
nasihatkon@kntu:code$ gcc struct5.c && ./a.out
First name: Ali
Last  name: Karimi
ID:  9612345
Age: 18
Gender: M

First name: Parvin
Last  name: Etesami
ID:  9612347
Age: 111
Gender: F
```

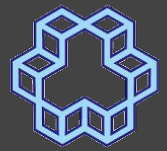
Is age a good record to keep?



Replace age with date of birth

1928
K. N. Toosi University of Technology

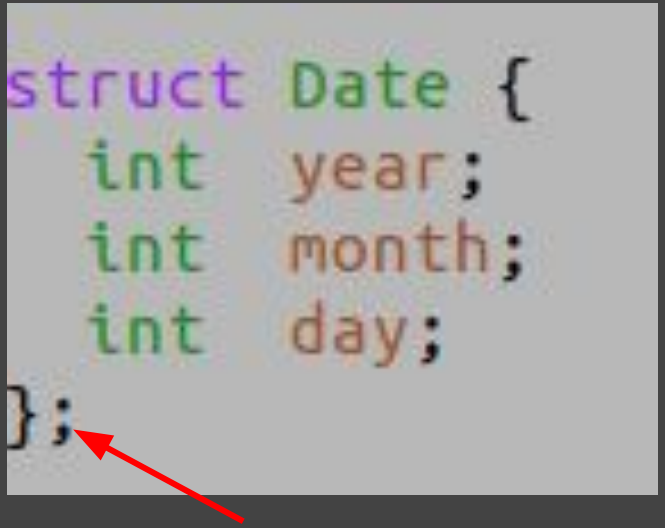
Replace age with date of birth




```
struct Date {
   int   year;
   int   month;
   int   day;
}
```

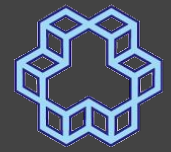
What's wrong?

Replace age with date of birth



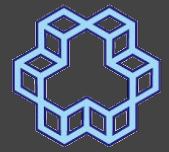
1928
K. N. Toosi University of Technology

Replace age with date of birth



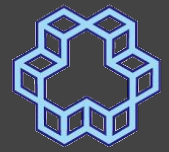

```c
struct Date {
  int  year;
  int  month;
  int  day;
};

struct Student {
  char firstName[20];
  char lastName[20];
  int id;
  struct Date DoB; // date of birth
  char gender;
};
```

Replace age with date of birth



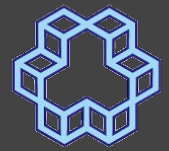

```
struct Student createStudent(char fName[], char lName[], int id,
                             int birth_year, int birth_month, int birth_day,
                             char gender) {
  struct Student s;

  strncpy(s.firstName, fName, 20);
  strncpy(s.lastName,  lName, 20);
  s.id  = id;
  s.DoB.year  = birth_year;
  s.DoB.month = birth_month;
  s.DoB.day   = birth_day;
  s.gender = gender;

  return s;
}
```

Replace age with date of birth



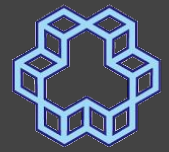

```
void printStudent(struct Student s) {
  printf("First name: %s\n", s.firstName);
  printf("Last  name: %s\n", s.lastName);
  printf("ID:  %7d\n", s.id);

  printf("DoB: %4d/%02d/%02d\n", s.DoB.year, s.DoB.month, s.DoB.day);

  printf("Gender: %c\n\n", s.gender);
}
```

Replace age with date of birth



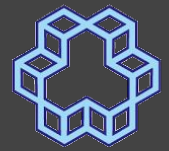

```
struct Student ali, parvin;

ali    = createStudent("Ali", "Karimi", 9612345, 1378,3,1, 'M');
parvin = createStudent("Parvin", "Etesami", 9612347, 1285, 12, 25, 'F');

printStudent(ali);
printStudent(parvin);
```

Replace age with date of birth



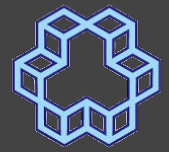

```
struct Student createStudent(char fName[], char lName[], int id,
                             int birth_year, int birth_month, int birth_day,
                             char gender) {
  struct Student s;

  strncpy(s.firstName, fName, 20);
  strncpy(s.lastName,  lName, 20);
  s.id  = id;
  s.DoB.year  = birth_year;
  s.DoB.month = birth_month;
  s.DoB.day   = birth_day;
  s.gender = gender;

  return s;
}
```

Replace age with date of birth



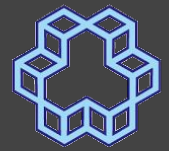

```c
void printStudent(struct Student s) {
  printf("First name: %s\n", s.firstName);
  printf("Last  name: %s\n", s.lastName);
  printf("ID:  %7d\n", s.id);

  printf("DoB: %4d/%02d/%02d\n", s.DoB.year, s.DoB.month, s.DoB.day);

  printf("Gender: %c\n\n", s.gender);
}
```

Replace age with date of birth




```
int main() {

  struct Student ali, parvin;

  ali    = createStudent("Ali", "Karimi", 9612345, 1378,3,1, 'M');
  parvin = createStudent("Parvin", "Etesami", 9612347, 1285, 12, 25, 'F');

  printStudent(ali);
  printStudent(parvin);

  return 0;
}
```

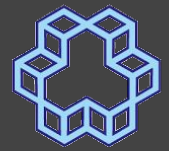
Replace age with date of birth




```c
int main() {

  struct Student ali, parvin;

  ali    = createStudent("Ali", "Karimi", 9612345, 1378,3,1, 'M');
  parvin = createStudent("Parvin", "Etesami", 9612347, 1285, 12, 25, 'F');

  printStudent(ali);
  printStudent(parvin);

  return 0;
}
```

```
nasihatkon@kntu:code$ gcc struct6.c && ./a.out
First name: Ali
Last  name: Karimi
ID:  9612345
DoB: 1378/03/01
Gender: M

First name: Parvin
Last  name: Etesami
ID:  9612347
DoB: 1285/12/25
Gender: F
```

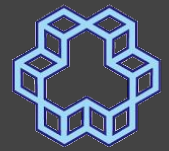
size of structures




```c
struct Date {
  int  year;
  int  month;
  int  day;
};

struct Student {
  char firstName[20];
  char lastName[20];
  int id;
  struct Date DoB; // date of birth
  char gender;
};
```

```c
printf("%lu\n", sizeof(struct Date));
printf("%lu\n", sizeof(struct Student));
```

size of structures

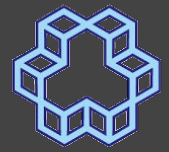



```c
struct Date {
  int  year;
  int  month;
  int  day;
};

struct Student {
  char firstName[20];
  char lastName[20];
  int id;
  struct Date DoB; // date of birth
  char gender;
};
```

```c
printf("%lu\n", sizeof(struct Date));
printf("%lu\n", sizeof(struct Student));
```

```
nasihatkon@kntu:code$ gcc struct7.c && ./a.out
12
60
```

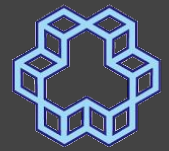
pointer to structures




```
ali    = createStudent("Ali", "Karimi", 9612345, 1378,3,1, 'M');
parvin = createStudent("Parvin", "Etesami", 9612347, 1285, 12, 25, 'F');

printStudent(&ali);
printStudent(&parvin);
```

```
void printStudent(struct Student *p) {
  printf("First name: %s\n", (*p).firstName);
  printf("Last  name: %s\n", (*p).lastName);
  printf("ID:  %7d\n", (*p).id);

  printf("DoB: %4d/%02d/%02d\n", (*p).DoB.year, (*p).DoB.month, (*p).DoB.day);

  printf("Gender: %c\n\n", (*p).gender);
}
```

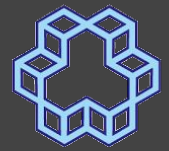
pointer to structures




```c
void printStudent(struct Student *p) {
  printf("First name: %s\n", (*p).firstName);
  printf("Last  name: %s\n", (*p).lastName);
  printf("ID:  %7d\n", (*p).id);

  printf("DoB: %4d/%02d/%02d\n", (*p).DoB.year, (*p).DoB.month, (*p).DoB.day);

  printf("Gender: %c\n\n", (*p).gender);
}
```

```c
void printStudent(struct Student *p) {
  printf("First name: %s\n", p->firstName);
  printf("Last  name: %s\n", p->lastName);
  printf("ID:  %7d\n", p->id);

  printf("DoB: %4d/%02d/%02d\n", p->DoB.year, p->DoB.month, p->DoB.day);

  printf("Gender: %c\n\n", p->gender);
}
```

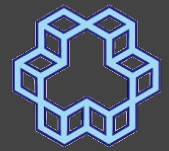
pointer to structures




```c
void printStudent(struct Student *p) {
  printf("First name: %s\n", (*p).firstName);
  printf("Last  name: %s\n", (*p).lastName);
  printf("ID:  %7d\n", (*p).id);

  printf("DoB: %4d/%02d/%02d\n", (*p).DoB.year, (*p).DoB.month, (*p).DoB.day);

  printf("Gender: %c\n\n", (*p).gender);
}
```

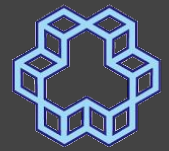
```c
void printStudent(struct Student *p) {
  printf("First name: %s\n", p->firstName);
  printf("Last  name: %s\n", p->lastName);
  printf("ID:  %7d\n", p->id);

  printf("DoB: %4d/%02d/%02d\n", p->DoB.year, p->DoB.month, p->DoB.day);

  printf("Gender: %c\n\n", p->gender);
}
```

pointer to structures




```c
struct Date {
  int  year;
  int  month;
  int  day;
};

struct Employee {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct  Employee supervisor;
};
```

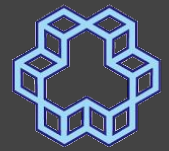
pointer to structures




```
struct Date {
  int  year;
  int  month;
  int  day;
};

struct Employee {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct  Employee supervisor;
};
```

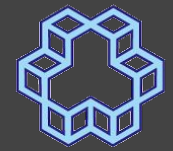
sizeof(struct Employee) = ?

pointer to structures



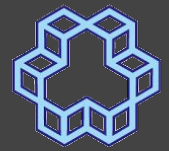

```c
struct Date {
  int  year;
  int  month;
  int  day;
};

struct Employee {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct Employee supervisor;
};
```

pointer to structures




```c
struct Date {
  int  year;
  int  month;
  int  day;
};

struct Employee {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct  Employee *supervisor;
};
```

pointer to structures



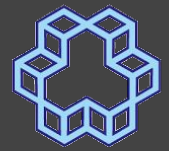

```c
struct Date {
  int  year;
  int  month;
  int  day;
};

struct Employee {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct  Employee *supervisor;
};
```

```c
printf("%lu\n", sizeof(struct Employee));
```

```
nasihatkon@kntu:code$ gcc struct10.5.c && ./a.out
64
```

64 bit (pointers are 8 bytes)

pointer to structures



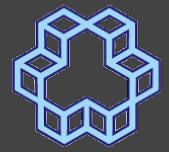

```c
struct Date {
  int  year;
  int  month;
  int  day;
};

struct Employee {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct  Employee *supervisor;
};
```

```c
printf("%lu\n", sizeof(struct Employee));
```

```
nasihatkon@kntu:code$ gcc struct10.5.c && ./a.out
64
```

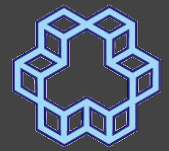
64 bit (pointers are 8 bytes)

pointer to structures



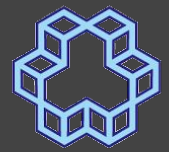

```c
struct Employee amin, behnam, parham, mahdi;

amin    = createEmployee("Amin", "Parchami", 1378,6,7, 'M');
behnam = createEmployee("Behnam", "Beigi", 1340, 12, 25, 'M');
parham = createEmployee("Parham", "Parviz", 1390, 12, 30, 'M');
mahdi = createEmployee("Mahdi", "Forozan", 1380, 10, 5, 'M');

amin.supervisor = NULL;
behnam.supervisor = &amin;
mahdi.supervisor = &amin;
parham.supervisor = &behnam;

printEmployee(&amin);
printEmployee(behnam.supervisor);
printEmployee(parham.supervisor->supervisor);
```

pointer to structures




```
struct Employee *amin, *behnam, *parham, *mahdi;
amin = &amin_s;
behnam = &behnam_s;
mahdi = &mahdi_s;
parham = &parham_s;

amin->supervisor = NULL;
behnam->supervisor = amin;
mahdi->supervisor  = amin;
parham->supervisor = behnam;

printEmployee(amin);
printEmployee(behnam->supervisor);
printEmployee(parham->supervisor);
printEmployee(parham->supervisor->supervisor);
```

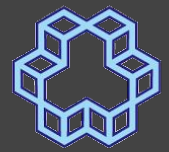
pointer to structures




```
struct Employee *amin, *behnam, *parh
amin = &amin_s;
behnam = &behnam_s;
mahdi = &mahdi_s;
parham = &parham_s;

amin->supervisor = NULL;
behnam->supervisor = amin;
mahdi->supervisor  = amin;
parham->supervisor = behnam;

printEmployee(amin);
printEmployee(behnam->supervisor);
printEmployee(parham->supervisor);
printEmployee(parham->supervisor->sup
```

```
nasihatkon@kntu:code$ gcc struct13.c && ./a.out
First name: Amin
Last  name: Parchami
DoB: 1378/06/07
Gender: M

First name: Amin
Last  name: Parchami
DoB: 1378/06/07
Gender: M

First name: Behnam
Last  name: Beigi
DoB: 1340/12/25
Gender: M

First name: Amin
Last  name: Parchami
DoB: 1378/06/07
Gender: M
```

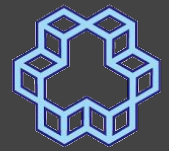
pointer to structures




```
struct Employee *amin, *behnam, *parh
amin = &amin_s;
behnam = &behnam_s;
mahdi = &mahdi_s;
parham = &parham_s;

amin->supervisor = NULL;
behnam->supervisor = amin;
mahdi->supervisor  = amin;
parham->supervisor = behnam;

printEmployee(amin);
printEmployee(behnam->supervisor);
printEmployee(parham->supervisor);
printEmployee(parham->supervisor->sup
```

```
nasihatkon@kntu:code$ gcc struct13.c && ./a.out
First name: Amin
Last  name: Parchami
DoB: 1378/06/07
Gender: M

First name: Amin
Last  name: Parchami
DoB: 1378/06/07
Gender: M

First name: Behnam
Last  name: Beigi
DoB: 1340/12/25
Gender: M

First name: Amin
Last  name: Parchami
DoB: 1378/06/07
Gender: M
```

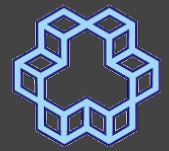
pointer to structures




```
amin->supervisor = NULL;
behnam->supervisor = amin;
mahdi->supervisor  = amin;
parham->supervisor = behnam;

printEmployee(findGodfather(parham));
```

```
struct Employee *findGodfather(struct Employee *pe) {

  while (pe->supervisor != NULL)
    pe = pe->supervisor;

  return pe;
}
```

pointer to structures




```
amin->supervisor = NULL;
behnam->supervisor = amin;
mahdi->supervisor  = amin;
parham->supervisor = behnam;

printEmployee(findGodfather(parham));
```

```
nasihatkon@kntu:code$ gcc struct14.c && ./a.out
First name: Amin
Last  name: Parchami
DoB: 1378/06/07
Gender: M
```

```
struct Employee *findGodfather(struct Employee *pe) {

  while (pe->supervisor != NULL)
    pe = pe->supervisor;

  return pe;
}
```

pointer to structures

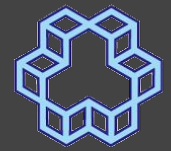
1928
K. N. Toosi University of Technology

```
amin->supervisor = NULL;
behnam->supervisor = amin;
mahdi->supervisor  = amin;
parham->supervisor = behnam;

printEmployee(findGodfather(parham));
```

```
nasihatkon@kntu:code$ gcc struct14.c && ./a.out
First name: Amin
Last  name: Parchami
DoB: 1378/06/07
Gender: M
```

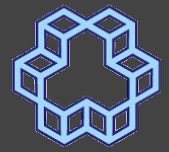
```
struct Employee *findGodfather(struct Employee *pe) {

  while (pe->supervisor != NULL)
    pe = pe->supervisor;

  return pe;
}
```

Other forms of declaration




```
struct Employee {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct  Employee *supervisor;

} ali, parvin;
```

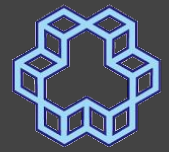
Other forms of declaration




```
struct Employee {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct  Employee *supervisor;

} ali, parvin;
```

```
struct {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct  Employee *supervisor;

} ali, parvin;
```

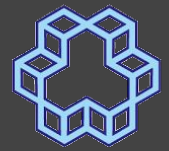
Other forms of declaration




```
struct Employee {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct  Employee *supervisor;

} ali, parvin;
```

```
struct {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct  Employee *supervisor;

} ali, parvin;
```

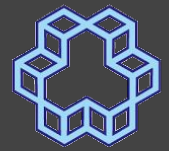
Other forms of declaration




```c
struct Date {
  int  year;
  int  month;
  int  day;
};

struct Employee {
  char firstName[20];
  char lastName[20];

  struct Date DoB; // date of birth

  char gender;
  struct  Employee *supervisor;

};
```

```c
struct Employee {
  char firstName[20];
  char lastName[20];

  struct {
    int  year;
    int  month;
    int  day;
  } DoB; // date of birth

  char gender;
  struct  Employee *supervisor;

};
```

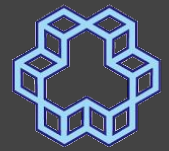
Structure initialization




```c
struct Employee {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct  Employee *supervisor;
};
```

```c
struct Employee amin = {"Amin", "Parchami", {1378, 10,6}, 'M', NULL};

printEmployee(&amin);
```

Structure initialization

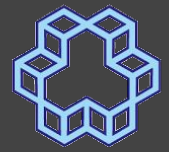



```c
struct Employee {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct  Employee *supervisor;
};
```

```
nasihatkon@kntu:code$ gcc struct19.c && ./a.out
First name: Amin
Last  name: Parchami
DoB: 1378/10/06
Gender: M
```

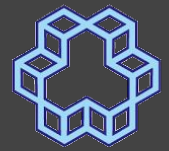
```c
struct Employee amin = {"Amin", "Parchami", {1378, 10,6}, 'M', NULL};

printEmployee(&amin);
```

Array of structures




```c
struct Employee amin = {"Amin", "Parchami", {1378, 10,6}, 'M', NULL};
struct Employee parvin = {"Parvin", "Etesami", {1278, 1,16}, 'F', NULL};
struct Employee ali = {"Ali", "Karimi", {1358, 1,6}, 'M', NULL};

struct Employee employees[3] = {amin, parvin, ali};

for (int i = 0; i < 3; i++)
  printEmployee(&employees[i]);
```

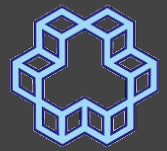
Array of structures




```c
struct Employee amin = {"Amin", "Parchami", {1378, 10,6}, 'M', NULL};
struct Employee parvin = {"Parvin", "Etesami", {1278, 1,16}, 'F', NULL};
struct Employee ali = {"Ali", "Karimi", {1358, 1,6}, 'M', NULL};

struct Employee employees[3] = {amin, parvin, ali};

for (int i = 0; i < 3; i++)
  printEmployee(&employees[i]);
```

```
nasihatkon@kntu:code$ gcc struct20.c && ./a.out
First name: Amin
Last  name: Parchami
DoB: 1378/10/06
Gender: M

First name: Parvin
Last  name: Etesami
DoB: 1278/01/16
Gender: F

First name: Ali
Last  name: Karimi
DoB: 1358/01/06
Gender: M
```

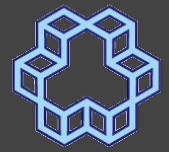
Array of structures




```c
struct Employee amin = {"Amin", "Parchami", {1378, 10,6}, 'M', NULL};
struct Employee parvin = {"Parvin", "Etesami", {1278, 1,16}, 'F', NULL};
struct Employee ali = {"Ali", "Karimi", {1358, 1,6}, 'M', NULL};

struct Employee employees[3] = {amin, parvin, ali};

for (int i = 0; i < 3; i++)
  printf("%s %s\n", employees[i].firstName, employees[i].lastName);
```

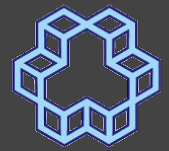
```
nasihatkon@kntu:code$ gcc struct21.c && ./a.out
Amin Parchami
Parvin Etesami
Ali Karimi
nasihatkon@kntu:code$
```

typedef




```
long unsigned int f(long unsigned int x)

int main() {
  long unsigned int a;
  long unsigned int b;
}

long unsigned int f(long unsigned int x) {
  return x*x;
}
```

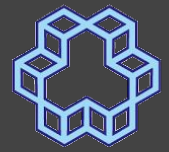
typedef




```
long unsigned int f(long unsigned int x)

int main() {
  long unsigned int a;
  long unsigned int b;
}

long unsigned int f(long unsigned int x) {
  return x*x;
}
```

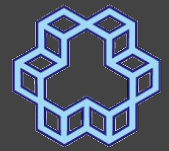
```
typedef long unsigned int uint64;

uint64 f(uint64 x)

int main() {
  uint64 a;
  uint64 b;
}

uint64 f(uint64 x) {
  return x*x;
}
```

typedef: Remember pointer to functions




```c
double f(double x) { return x*x-3*x+2;}

double g(double x) { return 1/x;}

double derivate(double (*h)(double), double x) {
  double delta = 1e-8;
  return ( h(x+delta) - h(x) ) / delta;
}
```

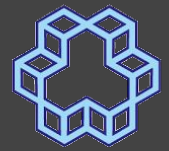
typedef




```
double f(double x) { return x*x-3*x+2;}

double g(double x) { return 1/x;}

double derivate(double (*h)(double), double x) {
  double delta = 1e-8;
  return ( h(x+delta) - h(x) ) / delta;
}
```

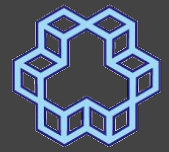
```
typedef double (* RealFunc)(double);

double derivate(RealFunc h, double x) {
  double delta = 1e-8;
  return ( h(x+delta) - h(x) ) / delta;
}
```

typedef




```c
struct employee {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct employee *supervisor;
};

typedef struct employee Employee;
```

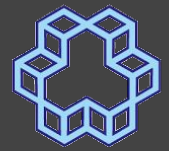
typedef




```c
struct employee {
  char firstName[20];
  char lastName[20];
  struct Date DoB; // date of birth
  char gender;
  struct employee *supervisor;
};

typedef struct employee Employee;
```

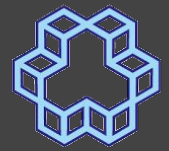
```c
Employee s;

strncpy(s.firstName, fName, 20);
strncpy(s.lastName,  lName, 20);
s.DoB.year   = birth_year;
```

typedef




```c
typedef struct {
  int  year;
  int  month;
  int  day;
} Date;

typedef struct employee {
  char firstName[20];
  char lastName[20];
  Date DoB; // date of birth
  char gender;
  struct employee *supervisor;
} Employee;
```

typedef




```c
typedef struct {
  int  year;
  int  month;
  int  day;
} Date;

typedef struct employee {
  char firstName[20];
  char lastName[20];
  Date DoB; // date of birth
  char gender;
  struct employee *supervisor;
} Employee;
```

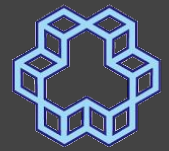
```c
void printEmployee(const Employee *p);
```

```c
Employee createEmployee(char fName[], char lName[],
                        int birth_year, int birth_m
                        char gender);
```

```c
Employee *findGodfather(Employee *pe) {
```

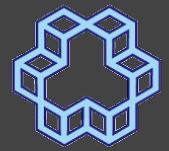
```c
Employee s;

strncpy(s.firstName, fName, 20);
strncpy(s.lastName,  lName, 20);
s.DoB.year  = birth_year;
s.DoB.month = birth_month;
s.DoB.day   = birth_day;
s.gender = gender;
```

Unions




```
struct Data {
  int  x;
  float y;
};

union UData {
  int  x;
  float y;
};
```

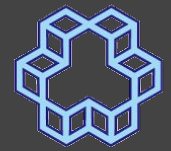
Unions




```c
struct Data {
  int  x;
  float y;
};

union UData {
  int  x;
  float y;
};
```

```c
int main() {

  struct Data  data;
  union  UData udata;

  printf("size(data)= %zu\n", sizeof(data));
  printf("size(udata)= %zu\n", sizeof(udata));

  return 0;
}
```

Unions

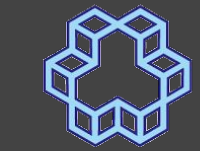



```c
struct Data {
  int  x;
  float y;
};

union UData {
  int  x;
  float y;
};
```

```c
int main() {

  struct Data  data;
  union  UData udata;

  printf("size(data)= %zu\n", sizeof(data));
  printf("size(udata)= %zu\n", sizeof(udata));

  return 0;
}
```

```
nasihatkon@kntu:code$ gcc union1.c && ./a.out
size(data)= 8
size(udata)= 4
```

Unions

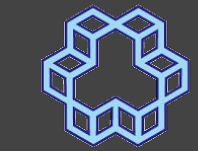



```c
union UData {
  int  x;
  float y;
};
```

```c
union  UData udata;

udata.x = 1;
printf("udata.x= %d\n",udata.x);

udata.y = 2.0;
printf("udata.y= %f\n",udata.y);
```

```c
union  UData udata;

udata.x = 1;
udata.y = 2.0;

printf("udata.x= %d\n",udata.x);
printf("udata.y= %f\n",udata.y);
```

Unions



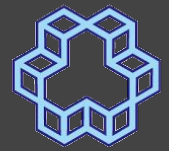

```c
union UData {
  int  x;
  float y;
};
```

```c
union  UData udata;

udata.x = 1;
printf("udata.x= %d\n",udata.x);

udata.y = 2.0;
printf("udata.y= %f\n",udata.y);
```

```c
union  UData udata;

udata.x = 1;
udata.y = 2.0;

printf("udata.x= %d\n",udata.x);
printf("udata.y= %f\n",udata.y);
```

```
nasihatkon@kntu:code$ gcc union2.c && ./a.out
udata.x= 1
udata.y= 2.000000
```

Unions




```c
union UData {
  int  x;
  float y;
};
```

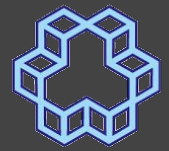
```c
union  UData udata;

udata.x = 1;
printf("udata.x= %d\n",udata.x);

udata.y = 2.0;
printf("udata.y= %f\n",udata.y);
```

```
nasihatkon@kntu:code$ gcc union2.c && ./a.out
udata.x= 1
udata.y= 2.000000
```

```c
union  UData udata;

udata.x = 1;
udata.y = 2.0;

printf("udata.x= %d\n",udata.x);
printf("udata.y= %f\n",udata.y);
```

```
nasihatkon@kntu:code$ gcc union3.c && ./a.out
udata.x= 1073741824
udata.y= 2.000000
```

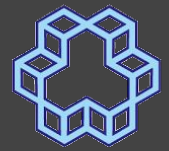
Unions




```c
#define INT    0
#define FLOAT 1

struct Number {
  int  type;

  union {
    int  i;
    float f;
  } value;

};
```

```c
struct Number createIntNumber(int val) {
  struct Number n;

  n.type = INT;
  n.value.i = val;

  return n;
}

struct Number createFloatNumber(float val) {
  struct Number n;

  n.type = FLOAT;
  n.value.f = val;

  return n;
}
```

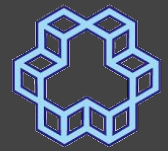
Unions




```c
#define INT   0
#define FLOAT 1

struct Number {
  int  type;

  union {
    int  i;
    float f;
  } value;

};
```

```c
void printNumber(struct Number n) {

  if (n.type == INT)
    printf("%d\n", n.value.i);
  else if (n.type == FLOAT)
    printf("%f\n", n.value.f);
}
```

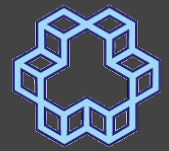
Unions




```c
#define INT    0
#define FLOAT 1

struct Number {
  int  type;

  union {
    int  i;
    float f;
  } value;

};
```

```c
void printNumber(struct Number n) {

  if (n.type == INT)
    printf("%d\n", n.value.i);
  else if (n.type == FLOAT)
    printf("%f\n", n.value.f);
}

int main() {

  struct Number n1, n2;

  n1 = createIntNumber(10);
  n2 = createFloatNumber(1.4);

  printNumber(n1);
  printNumber(n2);


  return 0;
}
```

Unions

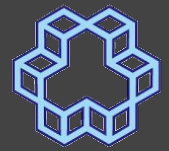



```c
#define INT    0
#define FLOAT 1

struct Number {
  int  type;

  union {
    int  i;
    float f;
  } value;

};
```

```c
void printNumber(struct Number n) {

  if (n.type == INT)
    printf("%d\n", n.value.i);
  else if (n.type == FLOAT)
    printf("%f\n", n.value.f);
}

int main() {

  struct Number n1, n2;

  n1 = createIntNumber(10);
  n2 = createFloatNumber(1.4);

  printNumber(n1);
  printNumber(n2);


  return 0;
}
```

```
nasihatkon@kntu:code$ gcc union4.c && ./a.out
10
1.400000
```

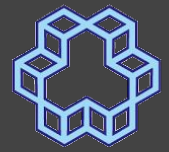
ENUM




```c
#define INT    0
#define FLOAT 1

struct Number {
  int  type;

  union {
    int  i;
    float f;
  } value;

};
```

```c
enum NumType {INT, FLOAT};

struct Number {
  enum NumType type;

  union {
    int  i;
    float f;
  } value;

};
```

Enum




```
enum Week {Shanbeh, Yekshanbeh, Doshanbeh, Seshanbeh,
           Chaharshanbeh, Panjshanbeh, Jome};
```

```
enum Week {Shanbeh=1, Yekshanbeh, Doshanbeh, Seshanbeh,
           Chaharshanbeh, Panjshanbeh, Jome};
```