Programming For Engineers

Practice 02 – Structure Data Type

by

Wan Azhar Wan Yusoff[1], Ahmad Fakhri Ab. Nasir[2]
Faculty of Manufacturing Engineering
wazhar@ump.edu.my[1], afakhri@ump.edu.my[2]

Practice 02 Information

- Competencies that you need to know about pointers: You must be able to:
  1. Define structure data type
  2. Define pointer structure data type.
  3. Define typedef structure data type.
  4. Access members of structure data type using dot operator.
  5. Access members of structure data type using arrow operator.
  6. Use nested structure data type.
  7. Use array of structure data type
  8. Pass structure data type to function and return a structure from a function.

Section A

- Sample Program Coding problem. You submit the code manually i.e. you write the solution on paper.

(1) We would like to create a structure data type for FKP student. We would like to store student name, ID and CPA. Write a program that create a student structure data type. Ali is a student with 3.14 CPA. His ID is 224. In your program display Ali information to the console.

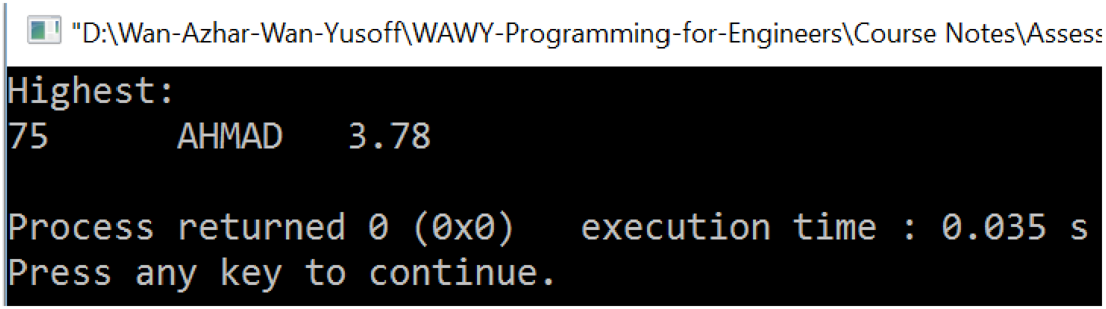
Section A

(2) Using the program that you write in Question (1), now create an array of student. Input the data for the three students listed below:

ID	Name	CPA
224	ALI	3.14
442	ABU	2.73
75	AHMAD	3.78

Extend your program that display the highest CPA first as shown below. You must you logical statement (if-else) to display the list.

Section A



"D:\Wan-Azhar-Wan-Yusoff\WAWY-Programming-for-Engineers\Course Notes\Assess

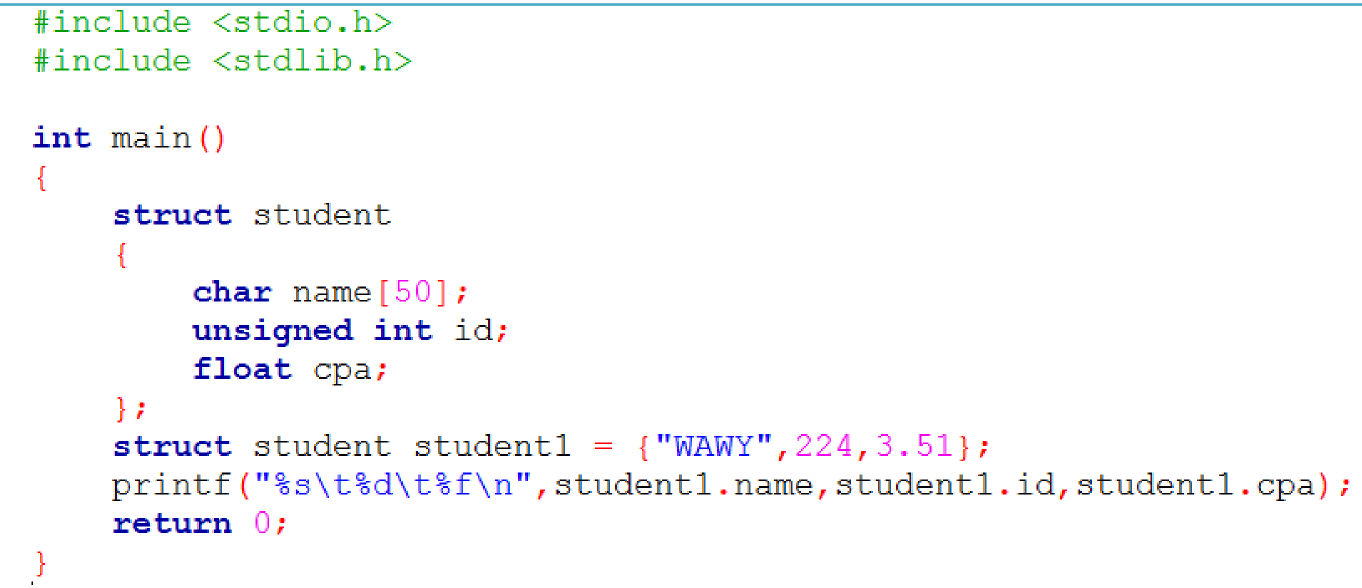
```
Highest:
75      AHMAD   3.78

Process returned 0 (0x0)   execution time : 0.035 s
Press any key to continue.
```

Section B

- Answer all questions by circling the correct output of the program.

1



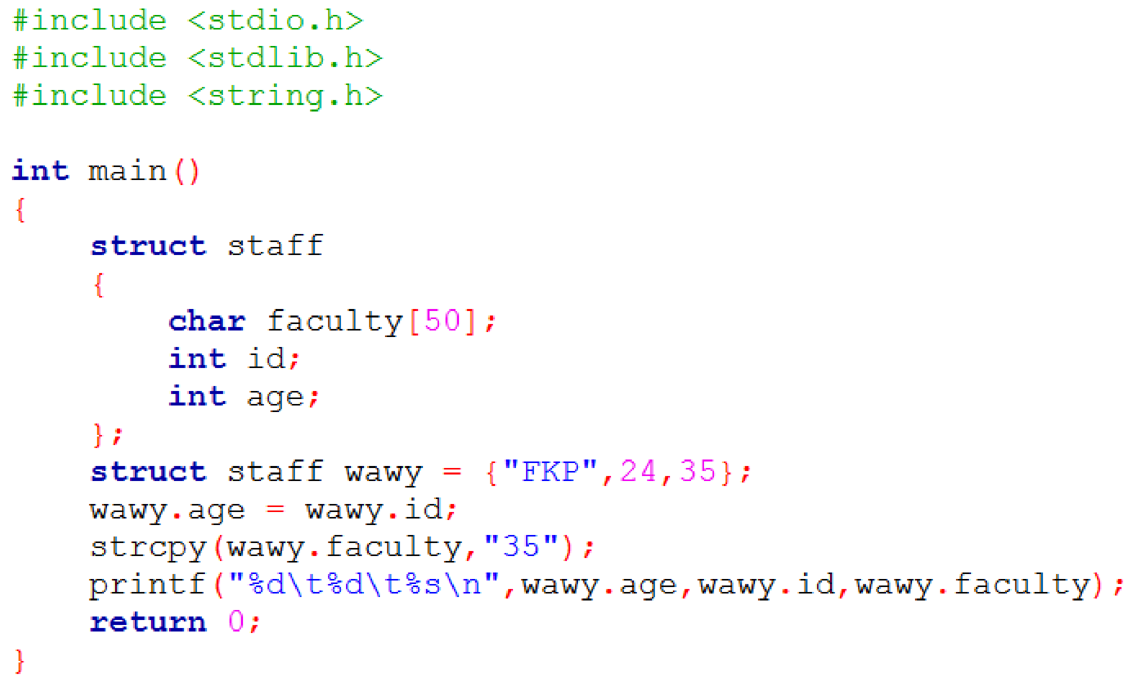
```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    struct student
    {
        char name[50];
        unsigned int id;
        float cpa;
    };
    struct student student1 = {"WAWY",224,3.51};
    printf("%s\t%d\t%f\n",student1.name,student1.id,student1.cpa);
    return 0;
}
```

(a)	224	3.510000	WAWY	(c)	WAWY	224	3.510000
(b)	WAWY	3.510000	224	(d)	224	WAWY	3.510000

Section B

2



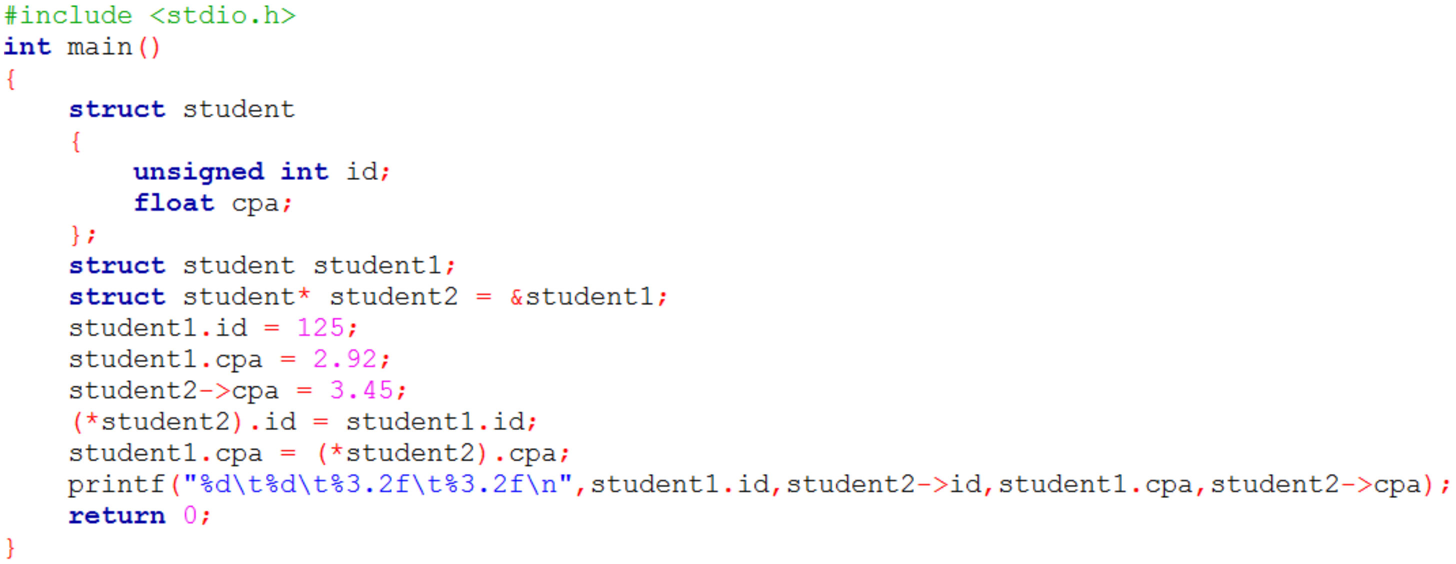
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    struct staff
    {
        char faculty[50];
        int id;
        int age;
    };
    struct staff wawy = {"FKP",24,35};
    wawy.age = wawy.id;
    strcpy(wawy.faculty,"35");
    printf("%d\t%d\t%s\n",wawy.age,wawy.id,wawy.faculty);
    return 0;
}
```

(a)	35 24 WAWY	(c)	24 24 24
(b)	24 24 35	(d)	35 24 FKP

Section B

3

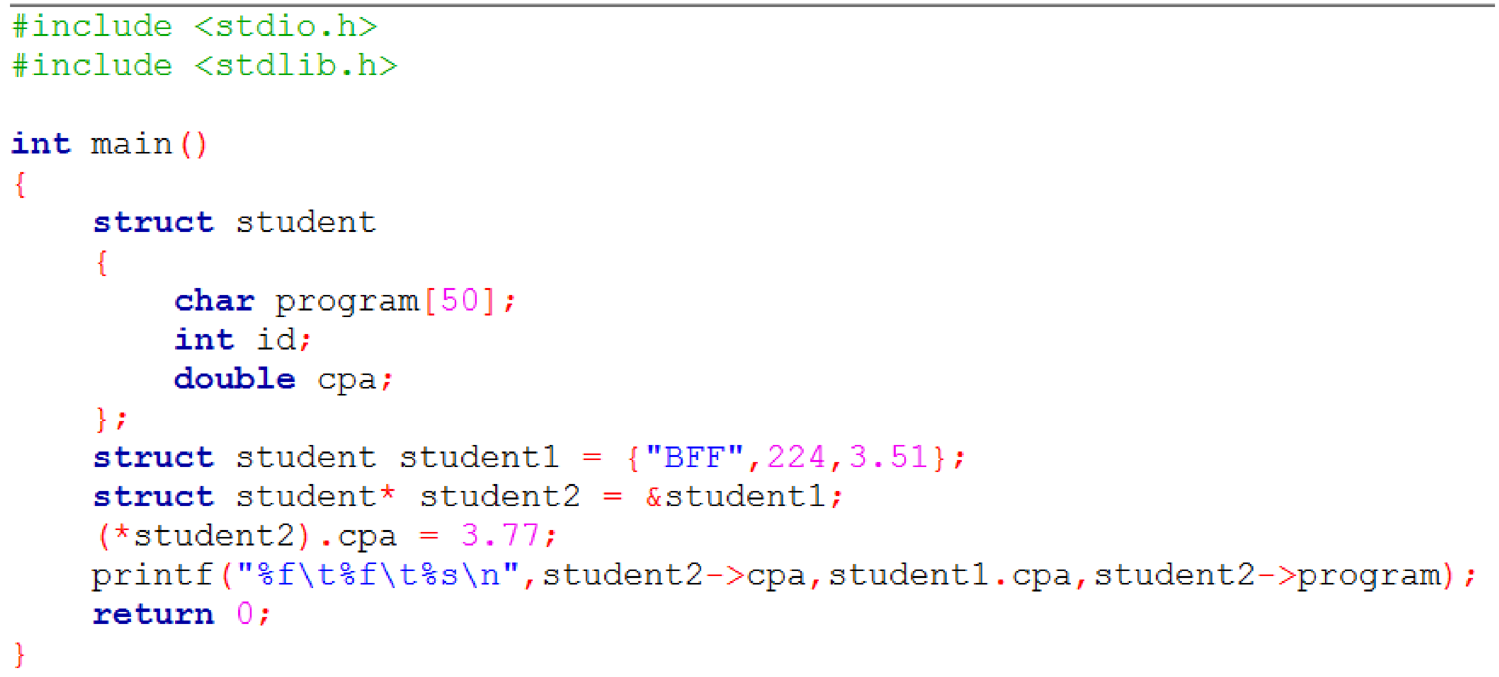


```
#include <stdio.h>
int main()
{
    struct student
    {
        unsigned int id;
        float cpa;
    };
    struct student student1;
    struct student* student2 = &student1;
    student1.id = 125;
    student1.cpa = 2.92;
    student2->cpa = 3.45;
    (*student2).id = student1.id;
    student1.cpa = (*student2).cpa;
    printf("%d\t%d\t%3.2f\t%3.2f\n",student1.id,student2->id,student1.cpa,student2->cpa);
    return 0;
}
```

(a)	3.45	125	3.45	3.45	(c)	125	125	2.92	2.92
(b)	125	125	3.45	3.45	(d)	2.92	2.92	3.45	3.45

Section B

4



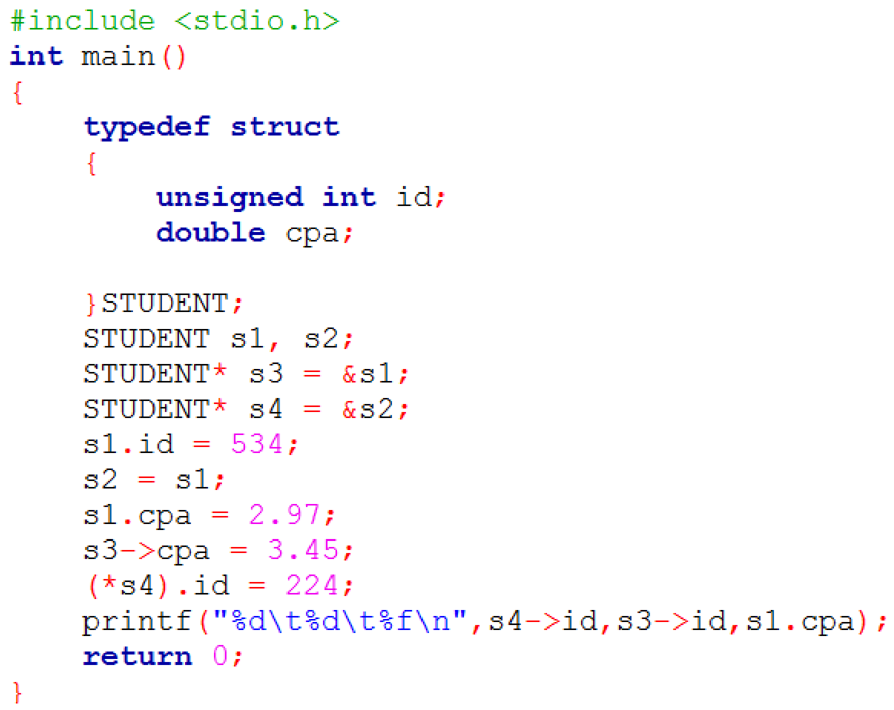
```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    struct student
    {
        char program[50];
        int id;
        double cpa;
    };
    struct student student1 = {"BFF",224,3.51};
    struct student* student2 = &student1;
    (*student2).cpa = 3.77;
    printf("%f\t%f\t%s\n",student2->cpa,student1.cpa,student2->program);
    return 0;
}
```

(a)	3.510000	3.770000	BFF	(c)	224	3.510000	BFF
(b)	3.510000	3.510000	BFF	(d)	3.770000	3.770000	BFF

Section B

5



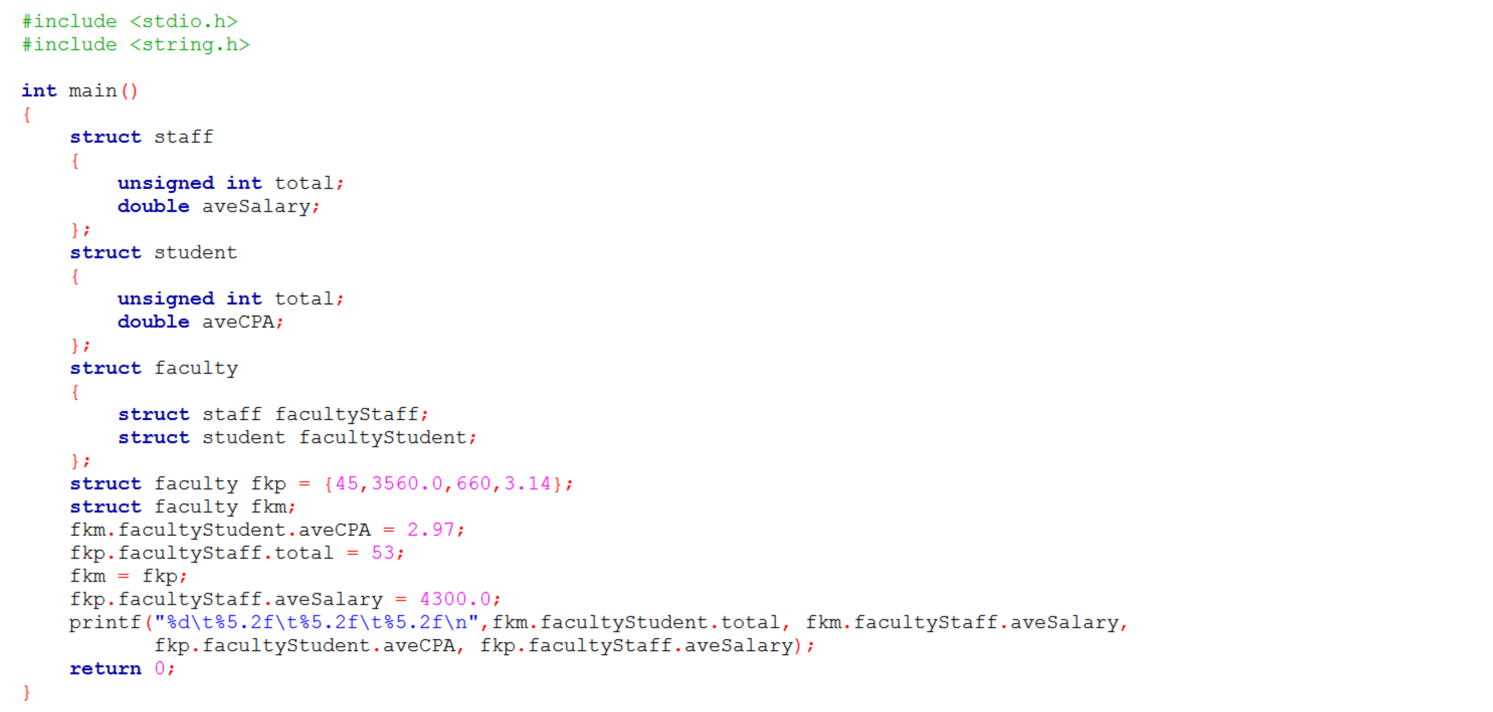
```
#include <stdio.h>
int main()
{
    typedef struct
    {
        unsigned int id;
        double cpa;

    }STUDENT;
    STUDENT s1, s2;
    STUDENT* s3 = &s1;
    STUDENT* s4 = &s2;
    s1.id = 534;
    s2 = s1;
    s1.cpa = 2.97;
    s3->cpa = 3.45;
    (*s4).id = 224;
    printf("%d\t%d\t%f\n",s4->id,s3->id,s1.cpa);
    return 0;
}
```

(a)	534	224	3.450000	(c)	224	224	2.970000
(b)	224	534	2.970000	(d)	224	534	3.450000

Section B

6



```c
#include <stdio.h>
#include <string.h>

int main()
{
    struct staff
    {
        unsigned int total;
        double aveSalary;
    };
    struct student
    {
        unsigned int total;
        double aveCPA;
    };
    struct faculty
    {
        struct staff facultyStaff;
        struct student facultyStudent;
    };
    struct faculty fkp = {45,3560.0,660,3.14};
    struct faculty fkm;
    fkm.facultyStudent.aveCPA = 2.97;
    fkp.facultyStaff.total = 53;
    fkm = fkp;
    fkp.facultyStaff.aveSalary = 4300.0;
    printf("%d\t%5.2f\t%5.2f\t%5.2f\n",fkm.facultyStudent.total, fkm.facultyStaff.aveSalary,
           fkp.facultyStudent.aveCPA, fkp.facultyStaff.aveSalary);
    return 0;
}
```

(a)	53 3560.00 3.143560.00	(c)	660 3560.00 2.974300.00
(b)	660 3560.00 2.973560.00	(d)	660 3560.00 3.144300.00

Section B

7

```c
#include <stdio.h>
struct student
{
    double cpa;
    double gpa[8];
};
void calculate_cpa(struct student*,int);
int main()
{
    struct student wawy = {0.0,
                           2.93,3.14,2.75, 3.34,
                           3.51,2.94,2.34,3.43};
    calculate_cpa(&wawy,8);
    printf("CPA: %f\n%f\t%f\n",wawy.cpa,wawy.gpa[2],wawy.gpa[3]);
    return 0;
}
void calculate_cpa(struct student* anyStudent, int semester)
{
    int i;
    double sum = 0.0;
    for (i=0;i<semester;i++)
    {
        sum = sum + anyStudent->gpa[i];
        if (anyStudent->gpa[i+1]>anyStudent->gpa[i] && (i<7))
            anyStudent->gpa[i]=anyStudent->gpa[i+1];
    }
    anyStudent->cpa = sum/semester;
}
```

(a)	CPA: 3.047500 2.940000 3.430000	(c)	CPA: 3.047500 2.750000 3.340000
(b)	CPA: 3.047500 3.340000 3.510000	(d)	CPA: 3.047500 3.140000 2.750000