VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Assignment – 3

CSE2010 - Advanced C Programming

Class Number: VL2020210504705 **Slot:** L43+L4 **Name :** Prashanth S **Roll :** 19MID0020

1. Write a C program to create and display a structure with members as employee Name, e Identification Number, Address, designation and salary.

Program flow

```
1  #define CLEAN_BUFFER while(getchar()!='\n');
2  #include<stdio.h>
3  #include<conio.h>
4
5  struct employee {
12
13 void print(struct employee obj1[],int n) {
24
25 void get_data(struct employee obj1[],int n) {
37
38 int main()
39 {
```

Code

```
1  #define CLEAN_BUFFER while(getchar()!='\n');
2  #include<stdio.h>
3  #include<conio.h>
4
5  struct employee {
6      int id;
7      char name[30];
8      char desig[30];
9      int salary;
10     char address[30];
11 };
12
13 void print(struct employee obj1[],int n) {
14   for(int i=0;i<n;i++) {
15       printf("Employee-%d details",i+1);
16       printf("\nID : %d",obj1[i].id);
17       printf("\nName : %s",obj1[i].name);
```

```
18          printf("\nDesignation : %s",obj1[i].desig);
19          printf("\nSalary : %d",obj1[i].salary);
20          printf("\nAddress : %s",obj1[i].address);
21          printf("\n\n");
22      }
23  }
24
25  void get_data(struct employee obj1[],int n) {
26      int id = 0;
27      for(int i=0;i<n;i++) {
28          printf("\nEnter Employee-%d details",i+1);
29          id+=1; obj1[i].id = id;
30          printf("\nEnter your name : ");       scanf("%d",&obj1[i].name);
31          CLEAN_BUFFER;
32          printf("Enter your designation : ");  scanf("%s",obj1[i].desig);
33          printf("Enter the Salary : ");          scanf("%d",&obj1[i].salary);
34           printf("Enter your address : ");       scanf("%s",obj1[i].address);
35      }
36  }
37
38  int main()
39  {
40              int n;
41              printf("Enter the number of employees : ");
42              scanf("%d",&n);
43              struct employee obj1[n];
44
45              get_data(obj1,n);
46              print(obj1,n);
47              return 0;
48  }
```

Output

```
Enter the number of employees : 3

Enter Employee-1 details
Enter your name : Prashanth
Enter your designation : Managing_Director
Enter the Salary : 40000
Enter your address : Chennai

Enter Employee-2 details
Enter your name : AndrewSams
Enter your designation : General_Manager
Enter the Salary : 30000
Enter your address : Bangalore

Enter Employee-3 details
Enter your name : MikeMothish
Enter your designation : Chief_executive_officer
Enter the Salary : 35000
Enter your address : Salem
Employee-1 details
ID : 1
Name :
Designation : Managing_Director
Salary : 40000
Address : Chennai

Employee-2 details
ID : 2
Name : 09Ü8ⁿ⌂
Designation : General_Manager
Salary : 30000
Address : Bangalore

Employee-3 details
ID : 3
Name : ⁿ⌂
Designation : Chief_executive_officer
Salary : 35000
Address : Salem


Process returned 0 (0x0)   execution time : 79.252 s
Press any key to continue.
```

2. With reference to the structure created in Q1, Write a C program to create record for N employees and print the details of the employee with greatest salary.

Program flow

```
 2  #include<stdio.h>
 3  #include<conio.h>
 4  #include<limits.h>
 5
 6  struct employee {
13
14  void print(struct employee obj1[],int n)  {
25
26  void salary_max(struct employee obj1[],int n)      {
39
40  void get_data(struct employee obj1[],int n) {
52
53  int main()
54  {
```

Code

```
 1  #define CLEAN_BUFFER while(getchar()!='\n');
 2  #include<stdio.h>
 3  #include<conio.h>
 4  #include<limits.h>
 5
 6  struct employee {
 7      int id;
 8      char name[30];
 9      char desig[30];
10      int salary;
11      char address[30];
12  };
13
```

```c
void print(struct employee obj1[],int n)   {
    for(int i=0;i<n;i++) {
          printf("Employee-details");
          printf("\nID : %d",obj1[i].id);
          printf("\nName : %s",obj1[i].name);
          printf("\nDesignation : %s",obj1[i].desig);
          printf("\nSalary : %d",obj1[i].salary);
          printf("\nAddress : %s",obj1[i].address);
          printf("\n\n");
    }
}

void salary_max(struct employee obj1[],int n)       {
        int maxi_salary = INT_MIN;
        int person = 0;

        for(int i=0;i<n;i++) {
        if ((maxi_salary)<obj1[i].salary)  {
             maxi_salary = obj1[i].salary;
             person = i;
           }
        }

        printf("\nMaximum salary of %d belongs to employee-%d : ",maxi_salary,(person+1));
}

void get_data(struct employee obj1[],int n) {
    int id = 0;
    for(int i=0;i<n;i++) {
          printf("\nEnter Employee-%d details",i+1);
          id+=1; obj1[i].id = id;
          printf("\nEnter your name : ");       scanf("%d",&obj1[i].name);
          CLEAN_BUFFER;
          printf("Enter your designation : ");  scanf("%s",obj1[i].desig);
          printf("Enter the Salary : ");          scanf("%d",&obj1[i].salary);
          printf("Enter your address : ");       scanf("%s",obj1[i].address);
    }
}
```

```c
int main()
{
        int n;
        printf("Enter the number of employees : ");
        scanf("%d",&n);
        struct employee obj1[n];
        get_data(obj1,n);
        salary_max(obj1,n);

        return 0;
}
```

Output

```
Enter the number of employees : 3

Enter Employee-1 details
Enter your name : Prashanth
Enter your designation : General_Manager
Enter the Salary : 80000
Enter your address : Chennai

Enter Employee-2 details
Enter your name : Maddymothish
Enter your designation : Managing_Director
Enter the Salary : 81000
Enter your address : Mumbai

Enter Employee-3 details
Enter your name : WillSmith
Enter your designation : CEO
Enter the Salary : 78000
Enter your address : Goa

Maximum salary of 81000 belongs to employee-2 :
Process returned 0 (0x0)   execution time : 64.795 s
Press any key to continue.
```

3. With reference to the Q2, Write a C program to do the same task by using pointers to array of structures.

Program flow

```
 2  #include<stdio.h>
 3  #include<conio.h>
 4  #include<limits.h>
 5
 6  struct employee {
13
14  void print(struct employee obj1[],int n)  {
25
26  void salary_max(struct employee obj1[],int n)      {
39
40  void get_data(struct employee obj1[],int n) {
52
53  int main()
54  {
```

Code

```
 1  #define CLEAN_BUFFER while(getchar()!='\n');
 2  #include<stdio.h>
 3  #include<conio.h>
 4  #include<limits.h>
 5
 6  struct employee {
 7      int id;
 8      char name[30];
 9      char desig[30];
10      int salary;
11      char address[30];
12  };
13
14  void print(struct employee obj1[],int n)  {
15    for(int i=0;i<n;i++) {
16        printf("Employee-details");
17        printf("\nID : %d",obj1[i].id);
```

```
18          printf("\nName : %s",obj1[i].name);
19          printf("\nDesignation : %s",obj1[i].desig);
20          printf("\nSalary : %d",obj1[i].salary);
21          printf("\nAddress : %s",obj1[i].address);
22          printf("\n\n");
23      }
24  }
25
26  void salary_max(struct employee *ptr,int n)       {
27          int maxi_salary = INT_MIN;
28          int person = 0;
29
30          for(int i=0;i<n;i++) {
31          if ((maxi_salary)<ptr->salary)  {
32                  maxi_salary = ptr->salary;
33              person = i;
34            }
35            ptr++;
36          }
37
38          printf("\nMaximum salary of %d belongs to employee-%d : ",maxi_salary,(person+1));
39  }
40
41  void get_data(struct employee obj1[],int n) {
42      int id = 0;
43      for(int i=0;i<n;i++) {
44          printf("\nEnter Employee-%d details",i+1);
45          id+=1; obj1[i].id = id;
46          printf("\nEnter your name : ");       scanf("%d",&obj1[i].name);
47          CLEAN_BUFFER;
```

```
47          CLEAN_BUFFER;
48          printf("Enter your designation : ");  scanf("%s",obj1[i].desig);
49          printf("Enter the Salary : ");        scanf("%d",&obj1[i].salary);
50          printf("Enter your address : ");      scanf("%s",obj1[i].address);
51      }
52  }
53
54  int main()
55  {
56          int n;
57          printf("Enter the number of employees : ");
58          scanf("%d",&n);
59          struct employee obj1[n];
60          get_data(obj1,n);
61          salary_max(obj1,n);
62          return 0;
63  }
64
```

Output

```
Enter the number of employees : 3

Enter Employee-1 details
Enter your name : Prashanth
Enter your designation : Managing_Director
Enter the Salary : 80000
Enter your address : Chennai

Enter Employee-2 details
Enter your name : MottuPatlu
Enter your designation : General_Manager
Enter the Salary : 77000
Enter your address : Mumbai

Enter Employee-3 details
Enter your name : Sushanth
Enter your designation : VicePresident
Enter the Salary : 88500
Enter your address : Pune

Maximum salary of 88500 belongs to employee-3 :
Process returned 0 (0x0)   execution time : 82.588 s
Press any key to continue.
```

4. Write a C program to define 3 structures as rectangle, square and circle and define their corresponding members. Pass the structure to 3 different functions to calculate its area and print the result.

Program flow

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

struct rectangle {   int length;    int breadth; }obj1;

struct square {   int side; }obj2;

struct circle   {  float radius;  }obj3;

int r_area(struct  rectangle obj) { if ((obj.length>0) && (obj.breadth>0))  { return obj.length*obj.breadth; } else { printf("Provic
int s_area(struct square obj)    { if (obj.side>0)    { return obj.side * obj.side; }                          else { printf("Provic
float c_area(struct circle obj)   { if (obj.radius>0) { return (3.14 * obj.radius * obj.radius); }             else { printf("Provic

int main() {
```

Code

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

struct rectangle {   int length;    int breadth; }obj1;

struct square {   int side; }obj2;

struct circle   {  float radius;  }obj3;

int r_area(struct  rectangle obj) { if ((obj.length>0) && (obj.breadth>0))  { return obj.length*obj.breadth; } else { printf("Pro
int s_area(struct square obj)    { if (obj.side>0)    { return obj.side * obj.side; }                          else { printf("Pro
float c_area(struct circle obj)   { if (obj.radius>0) { return (3.14 * obj.radius * obj.radius); }             else { printf("Pro

int main() {
   printf("1 : Rectangle");   printf("\n2 : Square");   printf("\n3 : Circle");
   int exp;
   printf("\nEnter your choice : "); scanf("%d",&exp);

   switch(exp) {

   case 1:
      printf("Enter the length and breadth : "); scanf("%d %d",&obj1.length,&obj1.breadth);
      printf("Rectangle area : %d",r_area(obj1));
      break;

   case 2:
      printf("Enter the side : ");  scanf("%d",&obj2.side);
      printf("Square area : %d",s_area(obj2));
      break;

   case 3:
      printf("Enter the radius : ");  scanf("%f",&obj3.radius);
      printf("Circle area : %f",c_area(obj3));
      break;

   }
   return 0;
}
```

Output

```
1 : Rectangle
2 : Square
3 : Circle
Enter your choice : 2
Enter the side : 10
Square area : 100
Process returned 0 (0x0)   execution time : 14.175 s
Press any key to continue.
```

```
1 : Rectangle
2 : Square
3 : Circle
Enter your choice : 1
Enter the length and breadth : 3 4
Rectangle area : 12
Process returned 0 (0x0)   execution time : 10.484 s
Press any key to continue.
```

```
1 : Rectangle
2 : Square
3 : Circle
Enter your choice : 2
Enter the side : -7
Provide positive values
Square area : 0
Process returned 0 (0x0)   execution time : 152.557 s
Press any key to continue.
```

5. Create a structure customer with members name, Id and amount. Also add appropriate functions to deposit, withdraw and check balance. Write a C program to create the structure and do the transaction based on user's choice. [Note: The amount has to be updated for every transaction.]

Program flow

```
1   #define CLEAN_BUFFER while(getchar()!='\n');
2   #include<stdio.h>
3   #include<conio.h>
4
5   struct bank {
10
11  int deposit(struct bank obj,int amt) {
15
16  int withdraw(struct bank obj , int amt) {
20
21  int check_balance(struct bank obj)  {     return obj.balance;  }
22
23  int main()  {
```

Code

```
1   #define CLEAN_BUFFER while(getchar()!='\n');
2   #include<stdio.h>
3   #include<conio.h>
4
5   struct bank {
6       int id;
7       char name[30];
8       int balance;
9   };
10
11  int deposit(struct bank obj,int amt) {
12          if (amt<0) { printf("Amount cannot be negative");  }
13          else {  obj.balance = obj.balance + amt;   }
14  }
15
```

```c
int withdraw(struct bank obj , int amt) {
   if (amt<obj.balance) {   obj.balance = obj.balance - amt;   }
   else {  printf("Amount exceeding the deposit"); }
}

int check_balance(struct bank obj)  {     return obj.balance;  }

int main()  {
      int n;
      printf("Enter the number of users : ");  scanf("%d",&n);

      struct bank obj[n];
      for(int i=0;i<n;i++) {
      printf("\nEnter the user_id : ");       scanf("%d",&obj[i].id);
      printf("Enter the user name : ");  scanf("%s",obj[i].name);
      printf("Enter the amount : ");      scanf("%d",&obj[i].balance);
      CLEAN_BUFFER;
      }

      printf("\n1: Deposit operation ");  printf("\n2: Withdrawn operation");  printf("\n3: Balance operation\n");

      for(int i=0;i<n;i++) {
            printf("\nUser - %d",i+1);

            int exp,amt;
            int start=1;
            while(start!=0) {
               printf("\nMention the operation you want to perform : "); scanf("%d",&exp);
               switch(exp) {
                  case 1: printf("Enter the amount to be deposited : "); scanf("%d",&amt);    obj[i].balance = deposit(obj[i],amt);
                  case 2: printf("Enter the amount to be withdrawn : ");  scanf("%d",&amt);  obj[i].balance = withdraw(obj[i],amt);
                  case 3: printf("Your balance amount is : %d",check_balance(obj[i])); break;
                  }
                  printf("\nEnter 1 to continue and 0 to stop ");   scanf("%d",&start);
            }
           printf("User- %d is terminated",i+1);    printf("\n");
      }
      return 0;
}
```

Output

```
Enter the number of users : 3

Enter the user_id : 1001
Enter the user name : Prashanth
Enter the amount : 5000

Enter the user_id : 1002
Enter the user name : Kumar
Enter the amount : 6000

Enter the user_id : 1003
Enter the user name : Mothish
Enter the amount : 7000

1: Deposit operation
2: Withdrawn operation
3: Balance operation

User - 1
Mention the operation you want to perform : 1
Enter the amount to be deposited : 4000

Enter 1 to continue and 0 to stop 1

Mention the operation you want to perform : 2
Enter the amount to be withdrawn : 100

Enter 1 to continue and 0 to stop 1

Mention the operation you want to perform : 3
Your balance amount is : 8900
Enter 1 to continue and 0 to stop 0
User- 1 is terminated

User - 2
Mention the operation you want to perform : 3
Your balance amount is : 6000
Enter 1 to continue and 0 to stop 0
User- 2 is terminated

User - 3
Mention the operation you want to perform : 2
Enter the amount to be withdrawn : 3500

Enter 1 to continue and 0 to stop 1

Enter 1 to continue and 0 to stop 1

Mention the operation you want to perform : 3
Your balance amount is : 3500
Enter 1 to continue and 0 to stop 0
User- 3 is terminated

Process returned 0 (0x0)   execution time : 6561.970 s
Press any key to continue.
```

6. Tell me about how efficient your programs are. (1 mark to show up your individuality, answer should be sound and sensible – Answers repeated no marks, so don't share your answers)

In the 7th question I used nested recursion, in which a stack within a stack of elements will be created and executed.

In 3rd question accessing the array of objects via pointer.

7. Just for learning: Write a C program to print numbers from 1 to 100 without using loop.

Program flow

```
1   #include<stdio.h>
2   #include<conio.h>
3
4   int fun1(int n) {
11
12  int main()  {
```

Code

```
1   #include<stdio.h>
2   #include<conio.h>
3
4   int fun1(int n) {
5      if (n>100) {  return ;   }
6      else {
7         if (n%10==1)  {  printf("\n%d   ",n);  fun1(n+1); }
8         else {    printf("%d    ",n);  fun1(n+1);   }
9      }
10  }
11
12  int main()  {
13            printf("The pattern is : ");
14            fun1(1);
15            return 0;
16  }
```

Output

```
The pattern is :
1   2    3    4    5    6    7    8    9    10
11   12    13    14    15    16    17    18    19    20
21   22    23    24    25    26    27    28    29    30
31   32    33    34    35    36    37    38    39    40
41   42    43    44    45    46    47    48    49    50
51   52    53    54    55    56    57    58    59    60
61   62    63    64    65    66    67    68    69    70
71   72    73    74    75    76    77    78    79    80
81   82    83    84    85    86    87    88    89    90
91   92    93    94    95    96    97    98    99    100
Process returned 0 (0x0)   execution time : 0.329 s
Press any key to continue.
```

Link to my code
https://github.com/PrashanthSingaravelan/WinterSemester-2021/tree/main/CSE2010%20Advanced%20C%20programming/Lab%20Assignments/Assignment-3