



SABARAGAMUWA UNIVERSITY OF SRI LANKA
FACULTY OF APPLIED SCIENCES
DEPARTMENT OF COMPUTING & INFORMATION SYSTEMS
BSc DEGREE PROGRAMME IN COMPUTING AND INFORMATION SYSTEMS

YEAR I SEMESTER I EXAMINATION OCTOBER/NOVEMBER 2021
ONLINE TAKE HOME EXAMINATIONS

IS 11302 – STRUCTURED PROGRAMMING TECHNIQUES

Start Date and Time : 09.10.2021 | 12.00 Noon

End Date and Time : 10.10.2021 | 12.00 Noon

INSTRUCTIONS TO CANDIDATES:

1. Please refer to the “Student Guide to Conducting Online Examination - Faculty of Applied Sciences” as the guidance document to set out your assignment.
2. You must complete the "Declaration by Student for Online Examination" sheet and upload to the given link - Declaration Form Submission.
3. Students should answer all the FOUR questions.
4. All assignments should comprise the standard Online Exam Cover Sheet given. No other front page will be accepted. A signed copy of the cover sheet must be attached to the front of the assignment before submission.
5. All assignments should be submitted to the link - Answer Submission. Answer file should be renamed/saved with candidate's index number.
6. Report Writing Guidelines:
 - a) Main Heading Font: Times New Roman – size 16
 - b) Sub Heading Font: Times New Roman – size 14
 - c) Body text Font: Times New Roman- size 12
 - d) Paragraph: Single line
 - e) Margins: Top – 1” Bottom – 1”. Left – 1”. Right – 1”
 - f) Header – include the Course code and Student Index Number on the Right-hand side
 - g) Footer – include the Page number on the Right-hand side

Que 1:

A)

1. While Loop

```
#include <stdio.h>
int main() {
    int i, sum = 0;
    i = 1;
    while (i <= 100) {
        sum += i;
        i=i+2;
    }
    printf("Sum = %d\n", sum);
    return 0;
}
```

2.Do While Loop

```
#include <stdio.h>
int main() {
    int i, sum = 0;
    i = 1;
    do{
        sum += i;
        i = i+2;
    }while (i<=100);
    printf("Sum = %d\n", sum);
    return 0;
}
```

3.For Loop

```
#include <stdio.h>
int main()
{
    int i, sum=0;
    for(i=1; i<=100; i+=2)
    {
        sum += i;
    }
    printf("Sum of odd numbers upto 100 is = %d\n", sum);
}
```

```

    return 0;
}

```

2)

```

#include<stdio.h>
#include<math.h>
int main()
{
    float SE,n1,n2,p;
    p=p0;
    n1=p1;
    n1=n2;

    float val;
    val=p*(1-p)*((1/n1)+(1/n2));
    SE=sqrt(val);
    if(a>=0)
    {
        printf("The value of SE :%f",SE);
    }
    else
    {
        printf("SE can not calculated its a negative value ");
    }
    return 0;
}

```

Que 2)

A)

Pass by value

In pass by value, the value of a function parameter is copied to another location of the memory. When accessing or modifying the variable within the function, it accesses only the copy. Thus, there is no effect on the original value.

Example:

```
#include<stdio.h>
void change(int num) {
    printf("Before adding value inside function num=%d \n",num);
    num=num+100;
    printf("After adding value inside function num=%d \n", num);
}
int main() {
    int x=100;
    printf("Before function call x=%d \n", x);
    change(x);//passing value in function
    printf("After function call x=%d \n", x);
    return 0;
}
```

Pass by references

In pass by reference, the memory address is passed to that function. In other words, the function gets access to the actual variable.

Example:

```
#include<stdio.h>
void change(int *num) {
    printf("Before adding value inside function num=%d \n",*num);
    (*num) += 100;
    printf("After adding value inside function num=%d \n", *num);
}
int main() {
    int x=100;
    printf("Before function call x=%d \n", x);
    change(&x);//passing reference in function
    printf("After function call x=%d \n", x);
    return 0;
}
```

Similarities and differences of pass by value and pass by reference.

Pass by value	Pass by references
A copy of value passes in to the function	The location of value passed to the function
The values of the actual parameters do not change by changing the formal parameters.	Here, actual parameter do change the by changing the formal parameter.

Actual and formal arguments are created at the different memory location	They are created in same memory location
--	--

B)

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

void student();
int total(int ,int ,int ,int , int );
float Average(float);

int main() {
for(int i=0;i<10;i++){
    student();
}
return 0;
}

void student(){
    int m1,m2,m3,m4,m5;
    printf("\nEnter the SPT Mark:");
    scanf("%d",&m1);
    printf("\nEnter the FIS Mark:");
    scanf("%d",&m2);
    printf("\nEnter the FOM Mark:");
    scanf("%d",&m3);
    printf("\nEnter the SAPT Mark:");
    scanf("%d",&m4);
    printf("\nEnter the GE Mark:");
    scanf("%d",&m5);

    printf("\nTotal:%d", total( m1,m2,m3,m4,m5));

    float avg=total( m1,m2,m3,m4,m5);

    printf("\nAverage:%0.2f",Average(avg));

}

int total(int m1,int m2,int m3,int m4, int m5){
    return m1,m2,m3,m4,m5;
```

```

}
float Average(float total){
    return total/5;
}

```

Que 3

Operating System Development

An Operating System is not a single program, but a assembly of software that exertion and communicate with each other. This is what I mean by "Operating Environment". Because Operating Systems are a group of software, in demand to develop an Operating System, one must know how to develop software. That is, one must know computer programming. One of the very robust reasons why C programming language is so prevalent and used so broadly is the flexibility of its use for memory management. Programmers have chances to switch how, when, and where to allocate and deallocate memory. Memory is allocated statically, automatically, or dynamically in C programming with the support of malloc and calloc functions.

Primarily, C was designed to implement the Unix operating system. Most of the Unix kernel, and all of its auxiliary tools and libraries, were written in C. Far along other folks found it useful for their programs without somewhat interference, and they began using it. Alternative solid purpose of using C programming language is that it sits close to operating system. This feature styles it an well-organized language because system level properties, such as memory, can be accessed with no trouble.

C is good choice for system-level programming. However, it is not perfect for all programmers and all circumstances. C pointers are a communal source of misunderstanding and programming errors. C also lacks explicit support for useful abstractions such as classes and objects. That may be a reason C is not desirable language for application-level programming. Innovative languages such as C++ and Java address these problems for application-level programs. C programming language is not inadequate to but used widely in operating systems, language compilers, network drivers, language interpreters, and system utilities areas of development. One way or another there are some drawbacks also in the C language.

Ineffective memory management is one of the key disadvantages when OS development in C. In C Language you don't need any memory management procedures since it automatically manages all assigned resources for you. But if you want to use dynamic allocation then you should allocate dynamically by malloc function. If you do this manually then you will get a subdivision fault error. Next, run time inspection. In the C programming language, the errors are not distinguished after each line of code. The compiler displays all the errors in the code only through the run-time of the whole which makes the checking of code (debugging) extremely complex in massive projects.

Likewise, the compiler doesn't check whether a variable was declared previously its use. The programmer must recall this rule while writing programs. Accordingly, C is greatest useful for

embedded systems, or applications that require the ability to be light-weight and have accurate control over system resources. C is lacking a lot of the functionality that more contemporary languages feature, but remains a core tool for Unix developers.

References.

<https://cs-fundamentals.com/tech-interview/c/why-we-use-c-programming-language#:~:text=Another%20strong%20reason%20of%20using,choice%20for%20system%20level%20programming.>

<https://dare2compete.com/blog/advantages-and-disadvantages-of-c-programming-language>

Embedded system development

C is a general-purpose computer programming language which can be used in inclusive variation of applications. Operating systems, application software for computers ranging from supercomputers to embedded systems are written in C. While C has been a multipurpose programming language, it has been the furthestmost appropriate one when it comes to Embedded Systems. In spite of being invented more than 30 years ago, when it comes to Embedded Systems, there is no additional programming language which even comes close to C. A computer system with a precise function within a larger system (mechanical or electrical system) is an embedded system. It is too probable that the Embedded Systems has real time computing restrictions, thereby making them as Real-Time-Systems (RTS). Today Embedded Systems are universally starting from wearable devices to high-speed transportation structures.

There are exact reasons for why is C the most favored language for Embedded Systems.

Processor self-determining

C language is not definite to any microprocessor/micro-controller or any system. It can work on numerous hardware configuration. C doesn't necessitate same set of hardware to run a program. C language is platform autonomous whereas the applications written in C are not. Hence the idea "Write once Compile anywhere" came.

Bit manipulation

C is more elastic, structured language that delivers "low level bit wise data manipulation" using the bit-wise operators. Using bit-wise operators, one could play with available bits, which comes handy when it comes to Embedded Systems (Ex: Handling registers).

Portability

Portability of a language is primarily vital when an application program must be moved to other operating systems. Code developed in C is more portable and user can compile it on other

platforms with least modifications. C code is effective, easy to comprehend, maintain and debug. A program can be written in one machine and can be run on further machines.

But in C in the ground of embedded system development, requires developers to appreciate and use technical coding techniques that can be complicated. Embedded Systems development has different set of challenges for developer as it is a resource constraint system. Similarly programming Embedded Systems includes dealing with actual hardware by controlling them via software.

Even though many languages occurred previously and many are still evolving, C has emerged as the number of language when it comes to Embedded Systems programming.

References

<https://www.emertxe.com/c-programming/why-is-c-the-most-preferred-language-for-embedded-systems/>

<https://www.qt.io/embedded-development-talk/embedded-software-programming-languages-pros-cons-and-comparisons-of-popular-languages>

Que 4

According to the scenario, Jayawardana and bro multinational national company willing to computerized their development. So, we have to input the employees' details to the system. So, in the beginning we have to create an structure for it. I have created a simple structure like in the figure below. The structure fully coded in C language. This is the simple preview of the system. We can input up to 100 employees.

This is the development project structure code:

```
#include<stdio.h>
struct employee
{
    int id,tel,day,month,year;
```



```

    char name[25] , *add[35];
}emp[100];
void main()
{
    int i,n;
    printf("Enter the no of employees\n");
    scanf("%d",&n);
    printf("Enter employee info as id , name , contact_number , address ,Joined_Day,
    Joined_Month, Joined_Year\n");
    for(i=0;i<n;i++)
    {
        scanf("%d %s %d %s %d %d",
        %d",&emp[i].id,emp[i].name,&emp[i].tel,emp[i].add,&emp[i].day,&emp[i].month,&emp[i].year
    );
    }
    printf("\nEMP_NAME\tEMP_NAME\tEMP_CONT_NUM\t\tAddress\t\t\tJoined_Day\tJoined_
    Month\tJoined_Year\n");
    for(i=0;i<n;i++)
    {
        printf("%d\t\t%s\t\t%d\t\t%s\t\t%d\t\t%d\t\t%d\n",emp[i].id,emp[i].name,emp[i].tel,emp[i].add,e
        mp[i].day,emp[i].month,emp[i].year);
    }
}

```

The code in C below:



```

1  #include<stdio.h>
2  struct employee
3  {
4      int id,tel,day,month,year;
5      char name[25] , *add[35];
6  }emp[100];
7
8  void main()
9  {
10     int i,n;
11     printf("Enter the no of employees\n");
12     scanf("%d",&n);
13     printf("Enter employee info as id , name , contact_number , address ,Joined_Day, Joined_Month, Joined_Year\n");
14     for(i=0;i<n;i++)
15     {
16         scanf("%d %s %d %s %d %d",&emp[i].id,emp[i].name,&emp[i].tel,emp[i].add,&emp[i].day,&emp[i].month,&emp[i].year);
17     }
18     printf("\nEMP_NAME\tEMP_NAME\tEMP_CONT_NUM\t\tAddress\t\t\tJoined_Day\tJoined_Month\tJoined_Year\n");
19     for(i=0;i<n;i++)
20     {
21         printf("%d\t\t%s\t\t%d\t\t%s\t\t%d\t\t%d\t\t%d\n",emp[i].id,emp[i].name,emp[i].tel,emp[i].add,emp[i].day,emp[i].month,emp[i].year);
22     }
23 }

```

```
"C:\Users\ASUS\Documents\sam.exe"
Enter the no of employees
2
Enter employee info as id , name , contact_number , address ,Joined_Day, Joined_Month, Joined_Year
23332
Raja
077898976
Peoples_Park,Colombo
21
03
2017
34321
Murali
076456744
Dam_Street,Colombo
04
11
2018

EMP_NAME      EMP_NAME      EMP_CONT_NUM      Address      Joined_Day      Joined_Month      Joined_Year
23332         Raja          77898976          Peoples_Park,Colombo      21              3              2017
34321         Murali        76456744          Dam_Street,Colombo      4              11             2018
Press any key to continue . . .
```

Fig 02

```
"C:\Users\ASUS\Documents\sam.exe"
Enter the no of employees
2
Enter employee info as id , name , contact_number , address ,Joined_Day, Joined_Month, Joined_Year
23332
Raja
077898976
Peoples_Park,Colombo
21
03
2017
34321
Murali
076456744
Dam_Street,Colombo
04
11
2018

EMP_NAME      EMP_NAME      EMP_CONT_NUM      Address      Joined_Day      Joined_Month      Joined_Year
23332         Raja          77898976          Peoples_Park,Colombo      21              3              2017
34321         Murali        76456744          Dam_Street,Colombo      4              11             2018
Press any key to continue . . .
```

Fig 03

Numbers of employees to input into the system

The entry should complete the following

We have to enter up to an employee Joined year for the completion of one entry