Victoria University of Bangladesh

Mid Term Assessment-Fall 2023

Name : MD Sujan ali

ID : 1119460091

Program : BBA

Batch : 46th

COURSE CODE : CSE-211

COURSE TITLE : Structure Programming Language

Submitted to : Md shahin khan shanto

1

Course title: Structured Programming Language.

Course code: CSE-211

Ans: to the Q.N (1)-(a)

(a) Define Structured Programming Language

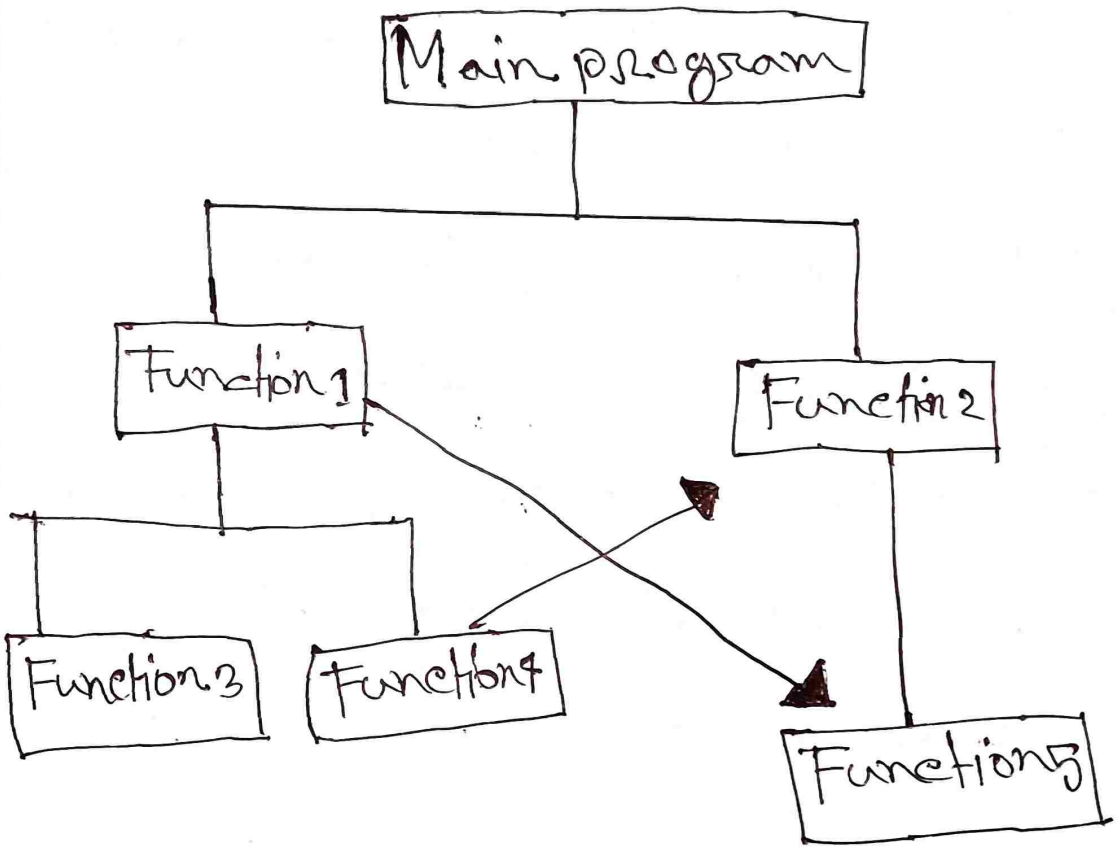
Structured Programming Language is a type of Programming Language that enforces a logical structure on the flow of control to make a program more readable and maintainable. It typically involves the use of block structures and modular design. It is a programming paradigm that facilitates the creation of Programs with readable code and reusable components. All modern Programming language support structured Progamming, but

P.t.o

The mechanisms to support like the syntax of the Programming language say when modules or elements of code can be reused from a library, it may also be possible to build structured code using modules written in different languages as long as they can obey a common module interface or Application program interface specification. when modules or elements of code can be reused it's possible to compromise data security and so vernance, so it's important to define and enforce a privacy policy controlling the use of modules that bring with them implicit data access rights.

P.t.o

Structured Programming language



p.t.o

Example //

Three examples of structured programming languages are:

C: known for its procedural programming features and structured syntax.

Pascal: Developed for teaching programming and emphasizes structured programming principles.

Ada: Designed for large-scale systems and enforces strong typing and modularization.

(1-b)

Ans: to the Q.N (1-b)

The types of Data types using in C.

Data is the primary need for any Programming language. This Data can be of various types, including integers, fractional numbers, characters, and symbols, or a group of entities. To handle these different types of Data, the concept of Data types has been introduced in C. Data types in C, signify the characteristic of the Data stored in a variable. Every type of Data has its own Data type that cannot be used to store other types of Data.

p.t.o

In C Programming the basic data types include: Primary data types in C are of 4 types, int, char, float and double. In this section, we are going to discuss all these data types in detail.

int: Integer data types for whole numbers.

float: floating-point data type for decimal numbers.

double: Double-precision floating-point data type.

char: character data type for single characters.

Bool: Boolean data type for true or false values.

P.T.O

short: Short integer data type.

long: Long integer data type.

long double: Extended precision floating-point data type.

These data types allow c programmers to work with different kinds of values in their programs.

Ans: to the Q.N (2-a)

(a) Variable in C Programming

C Programming a variable is a named storage location in memory used to store data. It has a specific data type, such as int, float, or char, which determines the type of data it can hold. Variables allows you to manipulate and store values in a Programming making it dynamic and capable of handling different inputs and scenarios. We can change the value of a variable in C or any other language, and we can also use it multiple time. We use Symbols in variables for representing the memory so that it become easily identi-

p.t.o

liable by any user.

Give a example

C programming a variable is a named storage location that holds a value, allowing you to manipulate and work with data in your program. Here's an example:-

```
#include <stdio.h>
int main() {
// Declaration and initialization of variables
int a = 5; // Integer variable
Float b = 3.14; //Floating-point variable
Char c = 'A'; //Character variable
```

p.t.o

```
// Displaying values
Printf("value of integer variable: %d\n",
        a);
Printf("value of float variable: %.2f\n",
        b);
Printf("value of character variable:
        %c\n", c);

return 0;
```

3

In this example, a, b, and c are variable of different type (int, float, char), and they store values that can be manipulated or used in various ways within the program.

p.t.o

Variables are containers for storing data values, like numbers and characters.

There are different types of variables defined with different keywords,

for example:

* int - stores integers (whole numbers) without decimals, such as 123 or -123
* float - stores floating point numbers, with decimal, such as 19.99 or -19.99
* char - stores single characters such as 'a' or 'B'. Char values are in surrounded by single quotes

Declaring, creating variables

To create a variable, specify the type and ~~assign~~ assign it a value:

p.t.o

Syntax

type variable name = value

where type is one of the types and variable name is the name of the variable (such as x or myname). The equal sign is used to assign a value to the variable.

So, to create a variable that should store a number, look at the following ~~example~~ example:

create a variable called mynum of type int and assign the value 15 to it.

```
int mynum = 15;
```

p.t.o

also declare a variable without assigning the value and assign the value later.

Example,

// Declare a variable

int myNum;

// Assign a value to the variable

myNum = 15;

The general rules for naming variable are:

* Name can contain letters, digits and underscores
* Name must begin with a letter or an underscore (_)
* Names are case sensitive (myVar and myvar are different variables)
* Names cannot contain whitespaces or special characters like !, #, %, etc.
* Reserved words (such as int) cannot be used as name.

Ans: to the Q.N (2-b)

(2-b)//

The rules of declaring variable in C Programming:

Syntax: Variable declaration typically Follow
The syntax: data_type variable_name;

Data type: Specify the data type of the variable (e.g. int, float, char) to define the kind of data it will hold.

Identifier Name: Choose a meaningful and descriptive name for the variable adhering to the rules for identifiers (e.g. cannot start with a number no spaces).

p.t.o

Initialization: optionally, you can initialize a variable at the time of declaration like int count = 0;
Uninitialized variables contain garbage values.

Scope: variables can have different scope (local or global) depending on where they are declared in the program

Storage class (optional): Specify the storage class of a variable using keywords like auto, static, extern, and register. If not specified, the default is usually auto.

NO Spaces: Do not use spaces within variable names.

p.t.o

One Declaration per Line: Declare one variable per line for better readability and maintainability.

Case Sensitivity: C is case-sensitive. So variable names like 'myVar' and 'myvar' are different.

Naming convention: Use meaningful names that reflect the variable's purpose. Start with a letter or underscore, followed by letters, digits, or underscores.

Memory Allocation: Memory is allocated for the variable based on its data type. The size and representation depend on the data type.

P.t.o

Variable name:

* Must begin with a letter (a-z, A-Z) or underscore (_).
* Subsequent characters can be letters, numbers, or underscores.

Numbers in Names: While you can use numbers in variable names, they cannot start with a number.

Example:

```
int myVariable; // valid declaration
float tempValue; // valid declaration
char user_name; // valid declaration
int 3count // Invalid: cannot start with a number.
```

These rules help create valid and meaningful variable names in C.

p.t.o

Multiple variables of the same type can be declared in a single line, separated by commas. Initialize variables at the time of declaration is possible variable must be declared before they are used in the program.

Example:

```
int main() {
    // Declaration and initialization
    int age = 25;
    // Declaration without initialization
    float Price;
    // variable with a global scope
    extern double global variable;

    return 0;
}
```

p.t.o

These rules help ensure proper usage and understanding variables in C Programming.

No Special characters (except underscore):

Only letters, digits and underscores are allowed in variable names.

Length Limitation: variable names should be of reasonable length maintain code readability.

Examples:-

```
int count;
float average_score;
char first_initial;
```

These rules ensure clarity and consistency in a program.

*white spaces are not allowed within variable name.

p.t.o

* A variable name should not exceed 31 characters.
* Declare variables before using them in the program.
* Each variable declaration should end with a semicolon (;).

Example:

```
int age; // valid declaration
float averagescore; // valid declaration
char initial; // valid declaration
int 2ndplace; // Invalid: starts with a digit
double my-score; // Invalid: Hyphen is not allowed
```

These rules ensure proper syntax and help in writing clean and understandable code.

Ans: to the Q.N (4.a)

(4-a) **Define loop and give an example**

A loop is a programming construct that enables the repetition of a set of instructions or statements. It allows a specific block of code to be executed repeatedly based on a condition or a predetermined number of iterations. Loops are essential for automating repetitive tasks in computer programs.

In computer programming, a loop is a sequence of instructions that is continually repeated until a certain condition is reached. Typically, a certain process is done, such as getting

p.t.o

an item of data and changing it, and then some condition is checked such as whether a counter has reached a prescribed number. If it hasn't the next instruction in the sequence is an instruction to return to the first instruction in the sequence and repeat the sequence. If the condition has been reached, the next instruction in sequence is brought to the next sequence of instruction or branches outside the loop. A loop is a fundamental programming idea that is commonly used in writing programs. An infinite loop is one that lacks a functioning exit routine. The result is that

p.t.o

that the loop repeats continually until the operating system senses it and terminates the Program with an error or until some other event occurs, such as having the Program automatically terminate after a certain duration or time.

Here's an example of a simple loop in Python using a while loop:

Python

```
counter = 0
while counter < 5:
    Print (counter)
    counter += 1
```

This loop prints numbers 0 through 4, incrementing the counter in each iteration until the conditional counter < 5 is no longer true.

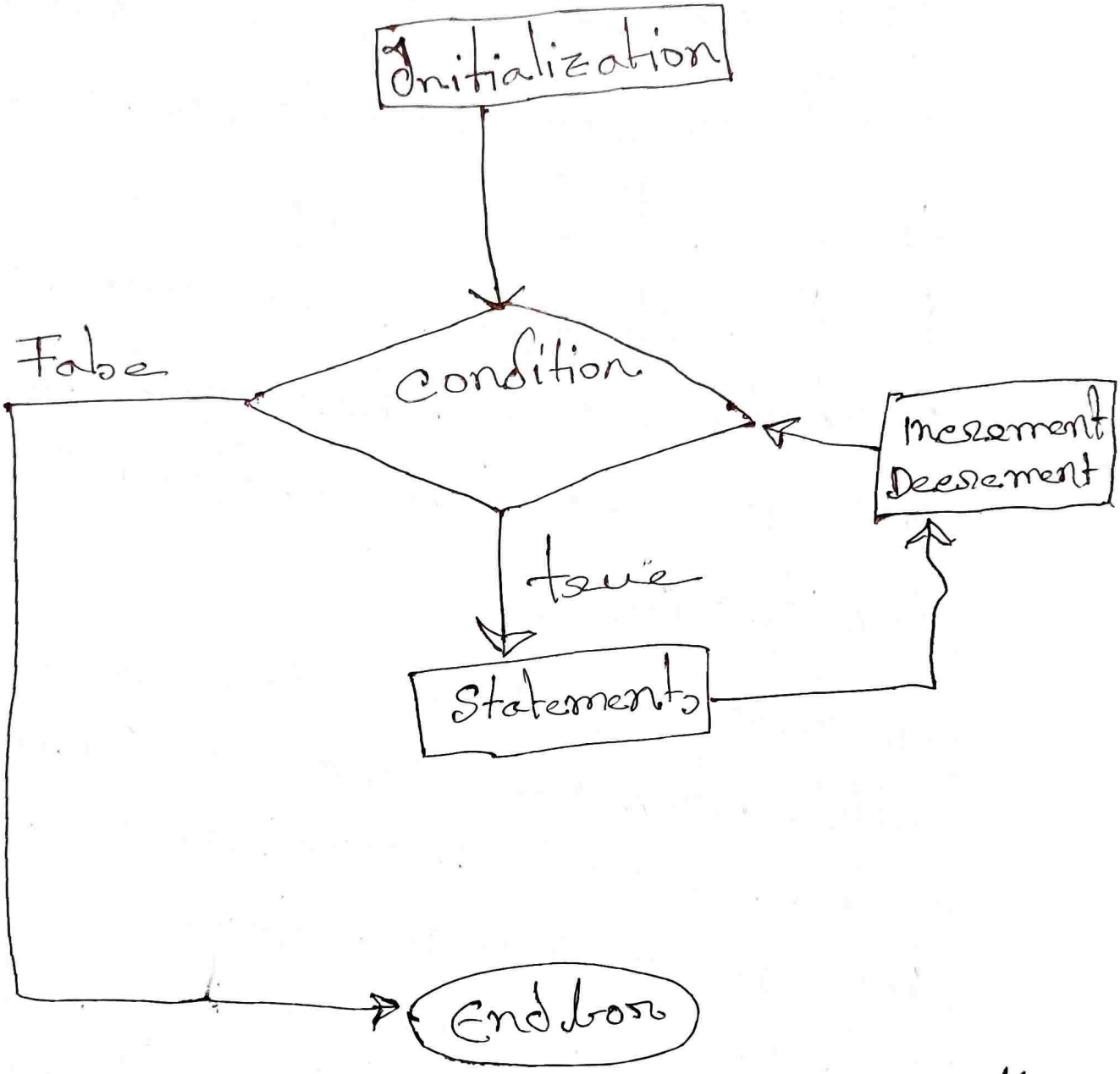
P.t.o

A "For" Loop is used to repeat a specific block of code a known number of times. For example, if we want to check the grade of every student in the class, we loop from 1 to that number. When the number of times is not known before hand we use a while loop.

A while loop is used to repeat a specific block of code an unknown number of time. until a condition is met. For example, if we want to ask a user for a number between 1 and 10, we don't know how many times the user may enter a larger number, so we kep asking while the number is not between 1 and 10.

p.t.o

For Loop in C Programming



For loops are used in C, as in other programming languages to run a block of code a specified number of times by evaluating condition as true or false.

Ans: to the Q.N (4-b)

(4-b)//

C- Program to print leap year:

First of all, it is important to know what is leap year? A, year has 365 days in a year, but a leap year has 366 days which comes after four year. are some points related to leap year.

* A leap year is a year, which is different than a normal year having 366 days instead of 365.
* A leap years comes once in four years, in which February month has 29 days. Within this additional day in February, a year becomes a leap year.

p.t.o

* Some leap years examples are- 1600, 1988, 1992, 1996, and 200.

* Although 1700, 1800, and 1900 are century years, not leap years.

1. year must be divisible by 4
2. year is divisible by 400 and not divisible by 100.

By putting these conditions in your code you can check year is a leap year or not. If the above conditions are satisfied. the year will be leap year these conditions can put with if-else or with && (and) and || (or).

Find leap year using c programming:

with the help of a c program. We will make easy to find a leap year.

P.t.o

~~Example~~

Here's a simple c program to check if a given year is a leap year.

```
#include<stdio.h>

main() {
 int year;
printf("Enter a year:");
scanf("%d", &year);
if ((((year%4==0) && ((year%400==0)
     ||(year%100!=0))

{
printf("%d is a leap year", &year);

} else {
printf("%d is not a leap year", &year);
}
 getch();
}
```

P.t.o

This program prompts the user to enter a year and then checks whether it's a leap year or not, based on the leap year rules.

e.g. example:

We will find the leap years between the range of two years like 2000 to 2020.

```
#include<stdio.h>
main() {
  int startyear, endyear, i;
printf("Enter a year to start searching
        the leap years:");
Scanf("%d", &startyear);
Printf("Enter a year to end the seach
        of leap year:");
```

P.t.o

```
scanf("%d",&endyear);
printf("Leap years between %d to %d are
        :\n",&startyear,&endyear);
for(i=startyear;i<=endyear;i++)
{
if((((i%4==0)&&((i%400==0)||(i%!=0))
{
printf("%d\n",&i);
}
}
getch();
}
```

Enter a year to start searching the leap year enter a year to end the search of leap years leap years between 1990 to 2020 are:
1992, 1996, 2000, 2004, 2008, 2012, 2016, 2020,