

1@ Data : Data are individual facts, statistics, or items of information; often numeric or values of ~~qualitative~~ qualitative or quantitative variables about one or more persons or objects: ~~slide~~ :

Information Information ~~are~~ is processed, organized and structured data. It provides context for data and enables decision making process.

1. (b) 2019-1(b)  
+ slide.

1(c) 2019-2(c)

2. (a) 2019-8(a)

2. (b) 2019-3(a)

3. (a) slide

(b) 2019-7(b)

(c) 2019-2(A)

4 @ slide

(b) slide

(c) Diagram (x)

5 @ slide

6a

RAM - Random access memory

ROM - Read Only memory

EPROM - Erasable programmable read only memory

EEPROM - Electrically erasable read-only memory

⑥ 2019 - 2③

⑦ 2019 - 8③

7✓✓ 8✓✓

**2018 Q/A Slon**

2018

1@

Data : Data are individual facts, statistics, or items of information; often numeric or values of ~~qualitative~~ qualitative or quantitative variables about one or more persons or objects: ~~abito~~ :

Information : Information ~~are~~ is processed, organized and structured data. It provides context for data and enables decision making process.

## What is Data?

Data is a collection of raw, unorganised facts and details like text, observations, figures, symbols and description of things etc. In other words, data does not carry any specific purpose and has no significance by itself. Moreover, data is measured in terms of bits and bytes – which are basic units of information in the context of computer storage and processing.

## What is Information?

Information is the processed, organised and structured data. It provides context for data and enables decision making. For example, a single customer's sale at a restaurant is data – this becomes information when the business is able to identify the most popular or least popular dish.

Data	Information
Data is unorganised and unrefined facts	Information comprises processed, organised data presented in a meaningful context
Data is an individual unit that contains raw materials which do not carry any specific meaning.	Information is a group of data that collectively carries a logical meaning.
Data doesn't depend on information.	Information depends on data.
It is measured in bits and bytes.	Information is measured in meaningful units like time, quantity, etc.
Raw data alone is insufficient for decision making	Information is sufficient for decision making
An example of data is a student's test score	The average score of a class is the information derived from the given data.

An operating system has three main functions:

- (1) manage the computer's resources, such as the central processing unit, memory, disk drives, and printers,
- (2) establish a user interface, and
- (3) execute and provide services for applications software.

## **Important functions of an operating System:**

**1 b(2019-1[b] ans + extra**

### **Security –**

The operating system uses password protection to protect user data and similar other techniques. it also prevents unauthorized access to programs and user data.

### **Control over system performance –**

Monitors overall system health to help improve performance. records the response time between service requests and system response to having a complete view of the system health. This can help improve performance by providing important information needed to troubleshoot problems.

### **Job accounting –**

Operating system Keeps track of time and resources used by various tasks and users, this information can be used to track resource usage for a particular user or group of users.

### **Error detecting aids –**

The operating system constantly monitors the system to detect errors and avoid the malfunctioning of a computer system.

### **Coordination between other software and users –**

Operating systems also coordinate and assign interpreters, compilers, assemblers, and other software to the various users of the computer systems.



**Memory Management –**

The OS maintains Primary Memory. The CPU may directly access main memory for quick storing. To run a program, it must first be loaded into main memory. An OS manages memory by doing the following: It maintains track of primary memory, i.e., which user programs utilize certain bytes. Memory addresses already allocated and unallocated. In multiprogramming, the OS selects which processes have memory access and for how long. It allocates memory to a process upon request and releases it when the process terminates or performs an I/O activity.

**Processor Management –**

The OS selects which processes get access to the CPU and how much processing time each gets. Process scheduling is an OS function. An OS manages processors by doing the following. Keeps track of process status. A traffic controller is a program that does this. Allocates a processor CPU to a process. When a process is finished, de-allocate processor.

**Device Management –**

It is the OS that manages device connectivity. It manages devices by doing the following tasks: Tracks all system devices. Every device has an Input/Output controller, which is programmed separately. Determines which processes have access to a device and how long. It efficiently assigns devices. Removes unused gadgets from the system.

**File Management –**

A file system is structured into directories for ease of use. These directories and files may contain others. An OS manages files in the following ways. It keeps track of data storage, user access, file status, and more... The file system encompasses all of these features.

- While loop checks the condition first and then executes the statement(s), whereas do while loop will execute the statement(s) at least once, then the condition is checked.
- While loop is entry controlled loop whereas do while is exit controlled loop.
- In the while loop, we do not need to add a semicolon at the end of a while condition but we need to add a semicolon at the end of the while condition in the do while loop.
- While loop statement(s) is executed zero times if the condition is false whereas do while statement is executed at least once.
- While loop allows initialization of counter variable before starting the body of a loop whereas do while loop allows initialization of counter variable before and after starting the body of a loop.

Here is an important difference between While and Do While Loop:

While	Do While
It checks the condition first and then executes statement(s)	This loop will execute the statement(s) at least once, then the condition is checked.
While loop allows initialization of counter variables before starting the body of a loop.	Do while loop allows initialization of counter variables before and after starting the body of a loop.
It is an entry controlled loop.	It is an exit controlled loop.
We do not need to add a semicolon at the end of a while condition.	We need to add a semicolon at the end of the while condition.
In case of a single statement, we do need to add brackets.	Brackets are always needed.
In this loop, the condition is mentioned at the starting of the loop.	The loop condition is specified after the block is executed.
Statement(s) can be executed zero times if the condition is false.	Statement is executed at least once.
Generally while loop is written as:	Generally do while loop is written as:

# Syntax of While loop

3 a

Here is a syntax of While loop:

```
while (condition) {  
    statements;  
}
```

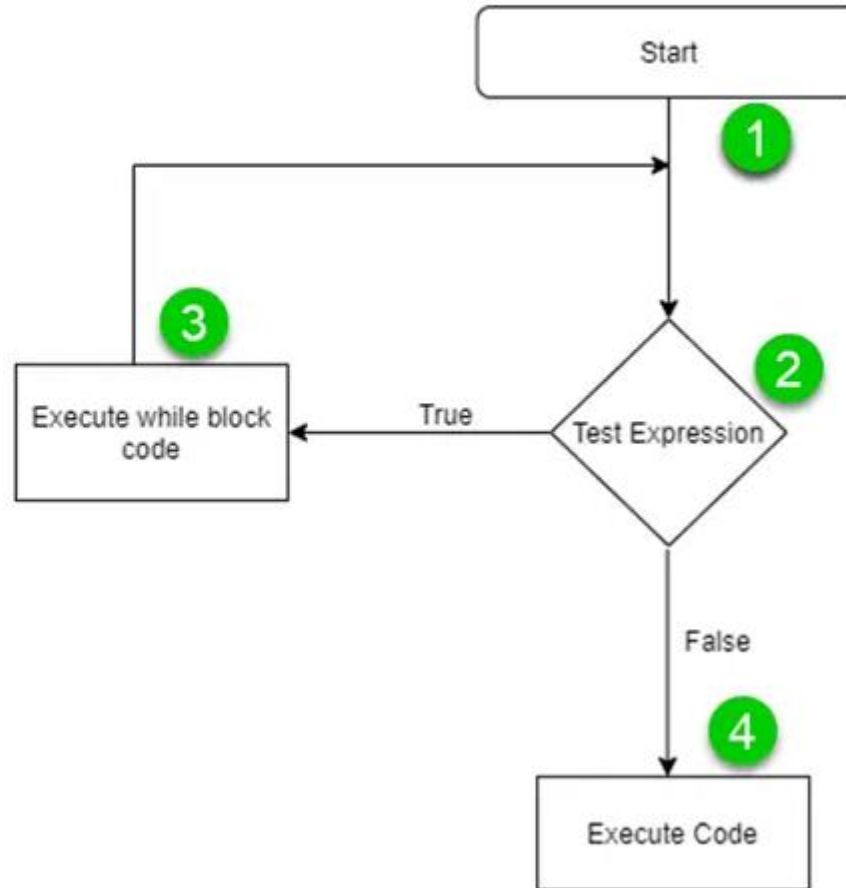
## Syntax Do While Loop

Here is a syntax of Do while loop:

```
do {  
    statements  
} while (expression);
```

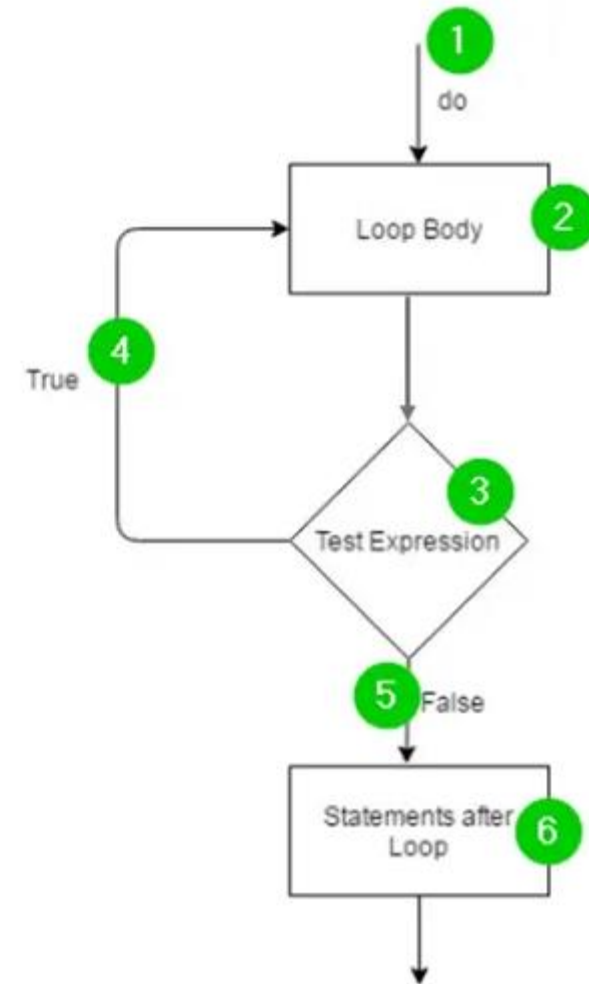
## How While Loop Works?

While loop works as follows:



## How Do-While Loop Works?

The Do-while loop works as follows:



## While Loop Example in C

3 a

Following program illustrates while loop in C programming with an example:

```
#include<stdio.h>
#include<conio.h>
int main()
{
int num=1;      //initializing the variable with value 1
while(num<=4)   //while loop with condition
{
printf("%d\n",num);
num++;          //incrementing operation
}
return 0;
}
```

Output:

```
1
2
3
4
```

## Do While Loop Example in C

The following program is a Do-while loop example to print a table of number 2 in C:

```
#include<stdio.h>
#include<conio.h>
int main()
{
int num=1;      //initializing the variable with value 1
do              //do-while loop
{
printf("%d\n",2*num);
num++;          //incrementing operation
} while(num<=4);
return 0;
}
```

Output:

```
2
4
6
8
```

# Structure vs class in C++

Difficulty Level : Easy • Last Updated : 26 Jan, 2022

In C++, a structure is the same as a class except for a few differences. The most important of them is **security**. A Structure is not secure and cannot hide its implementation details from the end-user while a class is secure and can hide its programming and designing details. **Following are some differences between a class and a structure.**

Class	Structure
Members of a class are private by default.	Members of a structure are public by default.
Memory allocation happens on the heap.	Memory allocation happens on a stack.
It is a reference type data type.	It is a value type data type.
It is declared using the <b>class</b> keyword.	It is declared using the <b>struct</b> keyword.

4 a

Some of the points that elaborate on these differences:

- 1) Members of a class are private by default and members of a structure are public by default.
- 2) Class is declared using the class keyword, and structure is declared using the struct keyword.
- 3) Inheritance is possible in class, and in structures.

**Class:** A class in C++ is the building block that leads to Object-Oriented programming. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A C++ class is like a blueprint for an object.

A Class is a user defined data-type which has data members and member functions. Data members are the data variables and member functions are the functions used to manipulate these variables and together these data members and member functions defines the properties and behavior of the objects in a Class.

The class is one of the defining ideas of object-oriented programming. Among the important ideas about classes are:

- A class can have subclasses that can inherit all or some of the characteristics of the class.
- In relation to each subclass, the class becomes the superclass. Subclasses can also define their own methods and variables that are not part of their superclass.
- The structure of a class and its subclasses is called the class hierarchy.



## Why Do Computers use Binary Numbers?

The binary numbering system is used in computing and electronics because it's the simplest counting method available. In addition, the binary numbering system is used to code everything from memory to images on the screen. Thus, it is the basis for the storage and transfer of data in most digital electronic devices.

The main reason the binary number system is used in computing is that it is simple. Computers don't understand language or numbers in the same way that we do. All they really have available to work with are switches and electrical signals, either on or off. To encode instructions or store values using switches – which can only be either off or on – the binary system is your obvious choice. In binary code, 'off' is represented by 0, and 'on' is represented by 1.

Computers use transistors to act as electronic switches. A small amount of current going into the transistor can generate a much higher output current: the smaller current switches on the higher current. If there's no current, the switch stays off. This is a fundamental explanation of how microchips work.

Values are stored in binary using these switches by setting them on (1) or off (0). One switch is equivalent to one bit, and so a bit also represents the smallest amount of information it is possible to configure. Eight switches – i.e., eight bits – make up a byte. Because each switch represents a line of digits in the binary counting system, eight switches represent any value between 0 and 256. Instructions are made up of strings of these bits, which the relevant hardware can read.

Nowadays, it's possible to fit millions of transistors on one microchip, but transistors had to be much larger in early computing. A counting system that uses more numbers would, arguably, allow for more values to be stored using much less space.