



# **Module 1**

## **Assignment**

### **IoT internship**

**Team Members: -**

**TL: Mahmoud Essam Fathy: 20221460231**  
**Ziad Ashraf Hafez Gaber: 20221374025**  
**Ziad Ashraf Ibrahim Taher: 20221369225**  
**Muhammed Ashraf: 20221372763**

# Module 1 Assignment – Part A – Problem 1

## < Compare Advantages and Disadvantages of IoT >

### < Answer >

Advantages of IoT	Disadvantages of IoT
Efficient resource utilization	Security
Minimize human effort	Privacy
Save time	Complexity
Enhance Data Collection	

#### Advantages

#### Detailed Points: -

- **Efficient resource utilization:** If we know the functionality and the way that how each device work, we definitely increase the efficient resource utilization as well as monitor natural resources
- **Minimize human effort:** As the devices of IoT interact and communicate with each other and do lot of tasks for us, then they minimize the human effort.
- **Save time:** As it reduces the human effort then it definitely saves out time. Time is the primary factor which can save through IoT platform.
- **Enhance Data Collection:** Easier in collecting data using sensors and no need to collect it manually.
- **Improve security:** Now, if we have a system that all these things are interconnected then we can make the system more secure and efficient.
- **Security:** As the IoT systems are interconnected and communicate over networks. The system offers little control despite any security measures, and it can be lead the various kinds of network attacks.
- **Privacy:** Even without the active participation on the user, the IoT system provides substantial personal data in maximum detail.
- **Complexity:** The designing, developing, and maintaining and enabling the large technology to IoT system is quite complicated.

#### Disadvantages

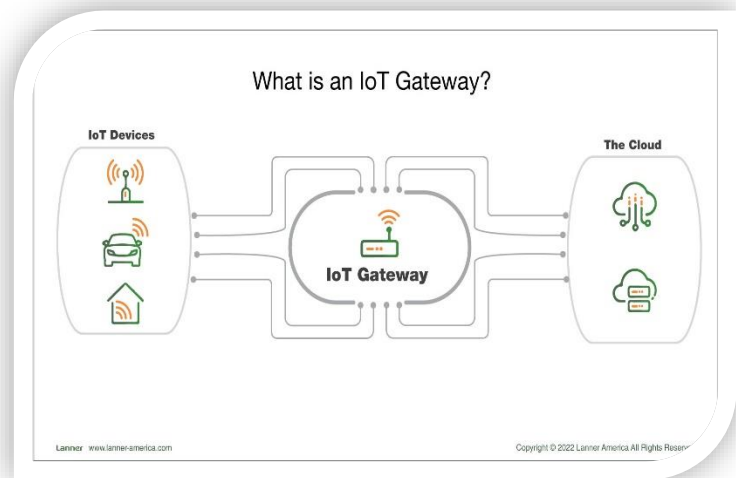


# Module 1 Assignment Part A – Problem 1

< Define what's the Gateway and explains its role in IoT layers >

< Answer >

**IoT Gateways:** are intelligent central hubs for Internet of Things (IoT) devices. That is, they connect IoT devices that have limited compute and storage to the cloud where data is managed and is used by end-user applications.



< Their use for main layers >

- **Sensor Layer:** At the sensor layer, devices collect data for future processing.  
<IoT devices operate at this layer>
- **Network or Data Acquisition Layer:** At this layer, data is aggregated from multiple sources and securely transmitted to processing systems
- **Data Pre-processing Layer:** At this layer, IoT sensor data undergoes pre-processing and basic data analytics to reduce data volume before it is transmitted on to cloud-based infrastructure.
- **Cloud Analysis or Application Layer:** Cloud-based infrastructure performs in-depth data analytics and provides applications and users with access to data and analytic results. Data storage or warehousing may also occur at this layer.

**Data acquisition systems (DAS)** are responsible for aggregating and converting data to a desired format.

# Module 1 Assignment – Part A – Problem 1

< Compare fog computing to cloud computing >  
< then comment what is better for the IoT Application? >

< Answer >

	Cloud Computing	fog computing
Delay	Has higher latency	Has lower latency
capacity	Doesn't reduction in data while sending	Reduce amount of data
Responsiveness	Is low	Is high
Security	has less Security	Has high security
Data Integration	Has multiple data sources can be integrated	Has multiple data sources and devices
Location of service	Services provided within the Internet	Services are provided at the edge of the local network
Communication mode	IP network	Wireless communication: WLAN, WiFi, 3G, 4G, ZigBee, etc. or wired communication (part of the IP networks)
Dependence on the quality of core network	Requires strong network core.	It can also work in a Weak network core
Speed	Access speed is high depending on the VM connectivity.	High even more compared to Cloud Computing.

**Cloud Computing** is better for applications that need massive data storage, high processing power, and complex data analysis. It's good for applications that can handle some network delays and want to save costs.

**Fog Computing** is better for applications that need real-time responsiveness, low latency, and high security. It's good for applications that need fast data processing and decision making.

The choice between Cloud and Fog Computing depends on the specific needs of the application. < In some cases, a combination of both can be the best option. >



# Module 1 Assignment Part A – Problem 1

< State the difference between HTTP and MQTT in IoT systems and which is preferred in most cases >

< Answer >

MQTT	HTTP
MQ Telemetry Transport, is a data transmission protocol. IBM developed it in 1999 for transmitting information between low-power sensors. It is based on the publish/subscribe model.	HTTP (Hyper Text Transfer Protocol) enables data retrieval on request. The protocol is a so-called "client-server" protocol. Thus, the HTTP protocol allows posting ("POST") data to a server (storing data on the server), which any other client (transferring stored data from a server to the client) can also request ("GET").
<b>MQTT is superior to HTTP</b> , if you have devices that communicate regularly. The MQTT protocol can keep a connection open for as long as possible, sending only a single data packet	HTTP communication, which requires you to open and close a connection (including TCP) for every data packet you want to send
MQTTs payload does not need to be encrypted or decrypted to be sent	HTTP document is always text-based; you must encrypt and decrypt any data format other than text
An MQTT packet contains at least two bytes	HTTP packet contains more than eight bytes

In summary, **MQTT is preferred for applications that require frequent and efficient communication between devices**, while **HTTP is preferred for applications that require a request-response model and larger data transmission**. The choice between MQTT and HTTP depends on factors such as **communication frequency, network conditions, security requirements, and integration with existing systems**.

# Module 1 Assignment – Part A – Problem 1

<Mention the three main kinds of program errors >  
<compare them to each other>

< Answer >

**Syntax errors:** When writing a program, the programmer should follow the proper standards and syntax of that language. Violating the grammar rules of the program can cause syntax errors. In other words, it is a mistake in the syntax of a sequence of characters or tokens when writing a program using a specific programming language.

**Runtime errors:** these are errors that are only produced as the program runs. These errors have code that compiles correctly and may run correctly under some situations but does something either unexpected or incorrect. Examples of run-time errors can be a user entering a String when the program expects a number, looping too far and getting an index out of bounds error.

**Logic errors:** Logic errors are different from the other error types because these errors do not cause any type of error from Java. Programs with logic errors will compile, run, and finish successfully, however they will not produce the correct results



# Module 1 Assignment Part A – Problem 1

## < State difference between source code and object code. >

### < Answer >

- **Source code:** Source code is programming statements that are created by a **programmer** with a text editor or a visual programming tool and then saved in a file.
- **Object code:** Object code generally refers to the output, **a compiled file**, which is produced when the Source Code is compiled with a suitable compiler to the programming language used to write source code into low level code which is understandable by machine.

SOURCE CODE	OBJECT CODE
Source code is human understandable.	Object code is not human understandable.
Source code is not directly understandable by machine.	Object code is machine understandable and executable.
It is written in a high-level language like C, C++, Java, Python, etc., or assembly language.	It is written in machine language through compiler or assembler or other translator.
It can be easily modified.	It can not be modified.
It contains comments for better understanding by programmer.	It does not contain comments for understanding by machine.
It contains a smaller number of statements than object code.	It contains a greater number of statements than source code.
It is less close. towards machine.	It is more close towards machine.
Performance of source code is less than object code as it is less close towards machine.	Performance of object code is more than source code as it is closer towards machine.
Source code is input to compiler or any other translator.	Object code is output of compiler or any other translator.
Source code is not system specific.	Object code is system specific.
It can be changed over time.	Source code needs to be compiled or translated by any other translator to get modified object code.
Language translators like compiler, assembler, interpreter is used to translate source code to object code.	Object code is machine code so it does not require any translation.
The source lines of code give the readability and understandability to the user. Use of fewer lines of code gives better performance by giving same results in most cases.	This is not the case with object code.

# Module 1 Assignment – Part A – Problem 1

## < What is Enum data type is used for >

### < Answer >

**An enumeration:** is a user-defined data type that consists of integral constants.

<To define an enumeration, keyword **Enum** is used>

**Example:**

```
<enum season { spring, summer, autumn, winter };>
```

1. name of Enum is: Season
2. Values of this Enum are "spring - summer - autumn - winter"

Next, declaration of each value in this Enum :

```
<enum season { spring = 0, summer = 4, autumn = 8, winter = 12 };>
```

**Why we use it ? look at next example :**

```
#include <iostream>
using namespace std;
enum week { Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday };
int main() {
    week today;
    today = Wednesday;
    cout << "Day " << today+1;
    return 0;
}
```

This code supposed to store all weeks in enum, and then print each value for each day

**NOTE :** you need to know that enum size is 4 bytes, that makes it only store integers.

"Remember, sunday here is considered as 0, and each value stored by its index, sunday is 0, Monday is 1 and etc"



# Module 1 Assignment – Part A – Problem 2, Freefall

## Pseudo Code

```
FUNCTION freefall(time)
return (acceleration * (time * time)) / 2)
END FUNCTION
DECLARE time as integer
DISPLAY "Please enter the time in seconds: "
READ time from user
COMPUTE distance=call freefall(time)
DISPLAY "You entered ", time, " seconds, so the distance is ", distance, " feet."
return 0
END FUNCTION
```

## Output value

```
Please enter the time in seconds: 150
You entered 150 in seconds,so the distance is 360000 feet.

Process returned 0 (0x0)   execution time : 2.493 s
Press any key to continue.
```



# Module 1 Assignment

## Part A – Problem 3

### C

```
Function Upper(c)
Display " " + "c" + " " + "c" + " " + "c"
Display " " + "c" + " " + "c"
END FUNCTION
Function lower(c)
Display " " + "c" + " " + "c"
Display " " + "c" + " " + "c" + " " + "c"
END FUNCTION
DECLARE c as CHARACTER
DISPLAY "Please put the character : "
READ c from user
Call UPPER(c)
for from 1 to 5
Display " " + "c"
END FOR
call LOWER(c)
```

Pseudo Code

```
Please put the character : M
  M M M
  M   M
M
M
M
M
M
  M   M
  M M M

Process returned 0 (0x0)   execution time : 1.779 s
Press any key to continue.
```

Output value

# Module 1 Assignment – Part A – Problem 3, velocity

## Pseudo Code

```
FUNCTION velocity_rule(Tc)
return 331.3+0.61*Tc
END FUNCTION
DECLARE start_temp,end_temp as integar
DISPLAY "Please put your start temp : "
READ start_temp from user
call setf(ios::fixed)
call precision(1)
  for i = start_temp to end_temp
DISPLAY "At " + i + " degrees Celsius the velocity of sound is " + velocity_rule(i) + " m/s\n"
END FOR
```

## Output value

```
Please put your start temp : 0
Please put your end temp : 5
At 0 degrees Celsius the velocity of sound is 331.3 m/s
At 1 degrees Celsius the velocity of sound is 331.9 m/s
At 2 degrees Celsius the velocity of sound is 332.5 m/s
At 3 degrees Celsius the velocity of sound is 333.1 m/s
At 4 degrees Celsius the velocity of sound is 333.7 m/s
At 5 degrees Celsius the velocity of sound is 334.4 m/s
```



# Module 1 Assignment

## Part A – Problem 4, interest due

```
//DECLARE account_balance, interest, total_with_interest, minimum_payment as unsigned integer
//DECLARE repeat as character
//do
//DISPLAY "Please enter your account balance: $"
// READ account_balance from user
//if account_balance <= 1000
// COMPUTE interest = round(account_balance * 0.015)
//else
//COMPUTE interest = round(1000 * 0.015 + (account_balance - 1000) * 0.01)
//END IF
//COMPUTE total_with_interest = account_balance + interest
//if total_with_interest <= 10
//minimum_payment = total_with_interest
//else
//COMPUTE minimum_payment = max(10, int(round(total_with_interest * 0.1)));
//END IF
//DISPLAY"Interest charged: $" + interest + "\n"
//DISPLAY "Total amount due with interest: $" + total_with_interest + "\n"
// DISPLAY"Minimum payment: $" + minimum_payment + "\n"
//DISPLAY "Do you want to calculate again? (Y/N) "
//READ repeat from user
//while repeat == 'Y' or repeat == 'y'
//END WHILE
```

Pseudo Code

```
Please enter your account balance: $500
Account balance: $500
Interest charged: $8
Total amount due with interest: $508
Minimum payment: $51
Do you want to calculate again? (Y/N) Y
Please enter your account balance: $700
Account balance: $700
Interest charged: $11
Total amount due with interest: $711
Minimum payment: $71
Do you want to calculate again? (Y/N) n
```

Output value

# Module 1 Assignment – Part A – Problem 5

## Some functions

Pseudo Code  
With output

```
//FUNCTION bool isPerfectSquare(int number)
//if number < 0
//return false
//END FUNCTION
//DECLARE x as int = sqrt(number)
//return x * x == number
//END
// FUNCTION reverseDigits(int number)
//DECLARE s as string = to_string(number)
//for i = s.length() from 0 to length of this data by -1
// DISPLAY s[i];
//END FUNCTION
//DISPLAY "\n";
//END
//FUNCTION int calculateSum(int input){
//DECLARE sum as int= 0;
//do
//COMPUTE sum = sum + input%10
//COMPUTE input = input/10;
//END
// while input!=0
//END while
//return sum

//DECLARE input as int
//DISPLAY "Enter an integer: "
// READ input from user
//if isPerfectSquare(input)
//DISPLAY"It's a perfect square."
//else
//DISPLAY "It's not a perfect square."
//END IF
//DISPLAY "The reverse of the input is: "
//call reverseDigits(input);
//DECLARE value as int= calculateSum(input);
//DISPLAY "The sum of digits is: " + value
//return success
```

```
Enter an integer: 4096
It's a perfect square.
The reverse of the input is: 6904
The sum of digits is: 19
```



# Module 1 Assignment

## Part A – Problem 6

### gallons

Define a constant variable LITER with a value 0.264172

Define a function milesPerGallon takes two inputs miles and liters and returns a double value, performing:

- Declare a double variable gallons
- Calculate gallons by multiplying LITER with lt
- Return the value of ml divided by gallons

END FUNCTION

Inside the main function, declare a char variable options

Start a do-while loop that continues while options is equal to 'r' or 'R'

Inside the loop, ask the user to enter the number of liters of gasoline and the number of miles traveled by the car

Call the milesPerGallon function with the entered values and display the result

Ask the user if they want to repeat the calculation or exit the program

If the user chooses to repeat, go back to step 5; otherwise, exit the program

Pseudo Code

```
Enter the number of Liters of gasoline:56
Enter the number of miles traveled by the car: 125000
Number of miles per gallon:8449.58
To repeat enter 'R' or enter 'E' to exit:R
Enter the number of Liters of gasoline:60
Enter the number of miles traveled by the car: 300000
Number of miles per gallon:18927.1
To repeat enter 'R' or enter 'E' to exit:E
```

Output value

# Module 1 Assignment – Part A – Problem 7

## Force question

Define the universal gravitational constant G which is equal to  $6.673 \times 10^{-8}$ .

Declare global variables m1, m2, and d.

Define the input function to set the values of m1, m2, and d by taking pointers to the variables as parameters and using cin to get input from the user.

Define the Force function to calculate the force between two objects by taking m1, m2, and d as parameters and returning the calculated force using the formula  $(G \cdot m1 \cdot m2) / (d \cdot d)$ .

Define the result function to display the results of the calculations by taking m1, m2, d, and the calculated force as parameters and using cout to display the values.

In the main function:

a. Call the input function to get the values of m1, m2, and d.

b. Calculate the force between the two objects by calling the Force function with the values of m1, m2, and d.

c. Display the results by calling the result function with the values of m1, m2, d, and the calculated force.

End the program.

Pseudo Code

Output value

```
what is mass of object 1 in grams?
700
what is mass of object 2 in grams?
1900
what is distance between the two objects in centimeters?
5000
Gravity G is 6.673e-08 N
mass of object 1 in grams is 700 G
mass of object 2 in grams is 1900 G
distance between the two objects in centimeters is 5000 CM
Force between the two objects in dynes is 3.55004e-09 dyens
where one dynes = ((g*cm)/(sec*sec))
```





# Module 1 Assignment Part A – Problem bonus PIZZA

```
making global constants as PI for 3.14, and converting from inches to CM with 2.54
declare function price_of_pizza with diameter_inch and price as inputs
  compute radius = diameter_inch / 2.0 * cm_per_inch
  compute area = pi*r^2
  return price per cm^2
END FUNCTION
read inches of small pizza and price from user
read inches of big pizza and price from user
compute price small , large
if(large<small)
  print(large is better)
else
  print(small is better)
end if
```

Pseudo Code

```
Enter diameter small pizza with(inches): 15
Enter price of small pizza: 20
Enter diameter large pizza with(inches): 30
Enter price of large pizza: 25
The value of small pizza price per CmSq :0.02, And for greater one : 0.01
Large pizza is a better .
```

Output value



# Module 1 Assignment – Part A – Problem Double bonus

## Bubble sort

```
function bubbleSort(arr, ssize)
  for each step in the data after performing 1 swap
    for each element in the array
      compare it with its next element
      if it is greater than the next element
        swap the elements
      end if
    end for
  end for
end function
```

```
function main()
  initialize an array of numbers
  calculate the size of the array
  sort the array using bubble sort
  print the sorted array
end function
```

Pseudo Code

Output value

```
Sorted Array in Ascending Order:
-9  -2  0  11  45
```

< Answer >