Peripheral: The devices that are external to the main processing function is called peripheral devices.

Example: keyboard, printer, input, output and storage device.

Q. CPU is a peripheral device, true or false?

Ans: Peripheral: The devices that are external to the main processing function is called peripheral devices.

CPU is the part of main processing function. So it is not peripheral device. That's mean the statement is false.

Peripheral devices connected via port & Interface to system bus like,

(SCSI) Small computer System Interface,

Integrated Drive Electronics (IDE),

personal computer Memory card International Association (PCMCIA)

Interfacing:

An interface is the point of interaction with software or computer hardware, or with peripheral devices.

There are 2 types of interfaces:

1. Hardware Interfaces
2. Software Interfaces

Hardware Interfaces: Hardware interfaces exist in computing system between many of the components such as the various buses, storage devices, other I/O devices.

Software Interface: A software interface may refer to a range of different types of interface at different levels.

Different levels of software interfacing:

1. An operating system may interface with pieces of hardware.
2. Applications or programs running on the operating system may need to interact via streams
3. In object oriented programs, objects within an application may need to interact via methods

Ports :

A port serves as an interface between the computer and other computers or peripheral devices.

Primarily there are two types of ports:

1. Physical Ports
2. Virtual Ports

physical ports :

Used for connecting a computer through a cable and a socket to a peripheral device.

1 Serial ports
2. USB ports
3. Parallel ports

Virtual ports:

Virtual ports are the data gates that allow software application to use hardware resources without any interference.

Primary Storage Devices:

primary memory: Conventional memory, immediate access by the CPU, Named as cache

Expanded storage: RAM, A buffer between cache memory and secondary memory

Secondary storage device:

It is a permanent memory, Non volatile memory. Data and programs must be copied to primary memory for CPU access. It is a electro-mechanical devices. Direct access storage devices are secondary storage device. It in both online and offline storge.

Online storage: Storage that is accessible to programs without human intervention

Both Primary and secondary storages are online storage.

Example: RAM, HDD

Offline storage: Storage that is not accessible to programs without human intervention

Example: CD, DVD, External HDD

Medium: That piece of technology which holds the data.

Access Time: The time to locate and read data

Transfer Rate:

Amount of data moves per seconds.

Serial port, Parallel port, USB:

serial port, Parallel port and USB ports are the physical ports that are used to connecting a computer through a cable and socket to a peripheral device.

Serial ports: A serial port is a type of interface that connects a device to the system unit by transmitting data only one bit at a time. It means, a serial port sends data bit by bit.

Parallel Ports: A parallel port is an interface that connects devices by transferring more than one bit at a time. The transmission speed of a parallel port is quite high as compared to a serial port.

Universal Serial Bus (USB): A USB port is a standard cable connection interface which can connect up to 127 different peripheral devices with a single connector type, greatly simplifying the process of attaching devices to a personal computer.

Relation between peripheral and interfacing:

peripheral means the devices that are external to the main processing function of the computer. classified as input, output, and storage devices.

Besides that, an interface is the point of interaction with software or computer hardware or with peripheral devices.

The peripheral device interacts with a computer system via interfacing points.

Class 2

What is Arduino:

Arduino board is an open source platform used to make electronics projects.

Basically, It is consists of both a microcontroller and a part of the software or Integrated Development Environment (IDE) that runs on your PC.

It is used to write & upload computer code to the physical board.

Arduino basic feature:

1. Arduino does not require a separate part of hardware.
2. In order to program a new code onto the board you can just use a USB cable
3. The Arduino IDE uses a basic version of c++, making it simpler to learn the program.
4. At last, Arduino board offers a typical form factor that breaks out the functions of the microcontroller into a more available package

Why Arduino:

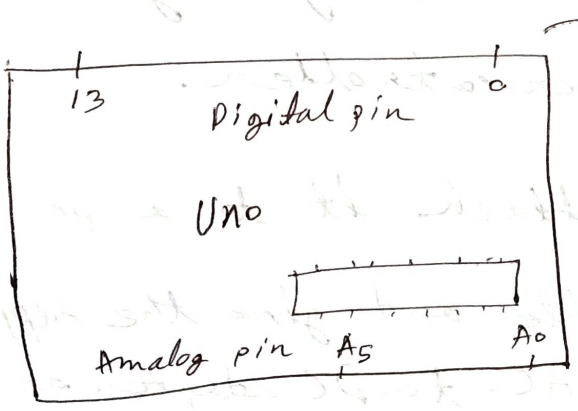
The Arduino software is very simple to use for beginners, yet flexible adequate for advanced users.

It runs on windows, Linux and Mac. Arduino also makes simpler the working process of microcontroller.

Beside that,
It is Inexpensive. It supports cross-platform. Simple, clear progeaming environment. It is extensible software & extensible hardware.

Arduino Uno → has processor → memory
↓ ↓
16 Mhz 2kB SRAM,
ATmega328 32kB flash

→ Digital I/O → Analog I/O
↓ ↓
14 6 input, 0 output



List of some Arduino :

1. Arduino Uno (R3)
2. LilyPad Arduino
3. Red Board
4. Arduino Mega(R3)
5. Arduino Leonardo.

Arduino Uno R3, Basic feature:

The Uno is a huge option for your initial Arduino.

It consists, 14-digital I/O pins, where 6-pins can be used as PWM (pulse width modulation outputs).

It also consists, 6-analog inputs, a reset button, a power jack, a USB connection and more.

Arduino uno includes everything required to hold up the microcontroller.

We can simply attach it to a pc with the help of a USB cable and give the supply to get started with a AC-to-DC adapter or battery.

List of project based on Arduino Uno:

1. Automatic Medicine Reminder Using Arduino
2. obstacle Avoiding Robot Using Arduino and Ultrasonic Sensor.
3. Google Assistant Based Voice controlled Home Automation Using DIY Arduino wifi shield.
4. Self balancing robot
5. Line follower.

Arduino Lilypad:

The Lilypad Arduino board is a wearable e-textile technology.

Arduino Shields:

Arduino shields are pre built circuit boards used to connect to a number of Arduino boards.

These shields fit on the top of the Arduino compatible boards. It provide an additional capabilities like comecting to the,
Internet, motor controlling, Providing wireless communication, LCD screen controlling

list of Arduino shields:

1. Ethernet shield: Allows you to connect your Arduino to the internet.

2. Relay shield: is a module with 4 mechanical relays that provides you an easy way to control high voltage.

3. ProtoShield:

The protoShield is a prototyping shield that makes it easy to prototype. It allows for easy connections between the breadboard and the Arduino.

4. Motor shield: This shield allows an easy control of motor direction and speed.

5. LCD shield: This shield makes it easy to use a 16x2 character LCD.

6. Capacitive Touchpad shield:

The Touchpad shield allows you to build simple capacitive touch interface.

7. Smoke Detector shield:

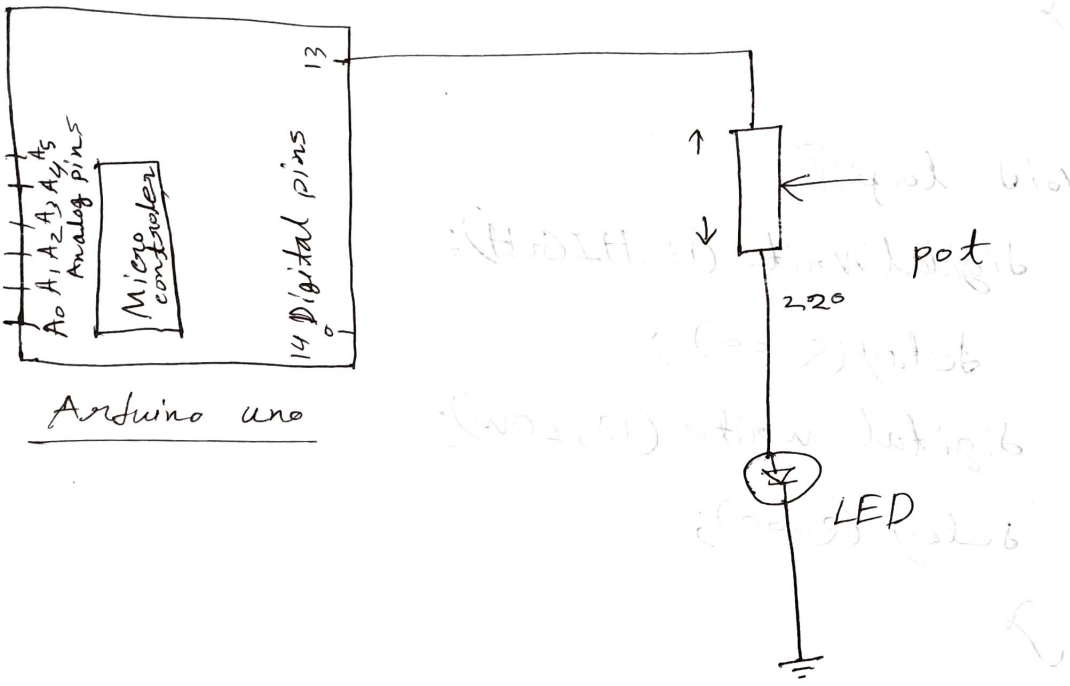
This shield can detect concentrations of combustible gas in the air and read it as an analogue value. Useful to make a smoke detector system.

8. 64-Button shield: with this shield you can connect up to 64 buttons to your Arduino.

9. Joystick shield kit: The joystick shield provides simple analog inputs and four separate buttons and one button under the joystick itself.

10. GSM/GPRS shield: Allows to connect your Arduino to GSM/GPRS cell phone network.

Circuit :- Light blink.



11. GPS Logger Shield
12. Wireless SD shield
13. cc300 wifi shield
14. HC-05 Bluetooth Shield
15. MP3 Player Shield.

Code : light blink

```
void setup(){
  pinMode(13, OUTPUT);
}

void loop(){
  digitalWrite(13, HIGH);
   delay(2000);
  digital write(13, LOW);
  delay(2000);
}
```

Class 3

Scope & Operations in Arduino

Scope: 3 types of scope in Arduino environment.

1. Global Scope.
2. Formal Scope.
3. Local Scope.

Example:

```
int i=2;                    → Global scope variable
void A(int, int);           → function prototype
void setup(){
Serial.begin(9600);
}
void loop(){
    int j=3;                → Local scope variable
    int k;
    k=A(i,j);
    Serial.println(k);
    delay(500);
}
```

```
                            → formal scope variable
int A(int x, int y){
    int result;
    result = x*y
    return result;
}
```

Operator: There are five types of operator in Arduino,

1. Arithmetic Operator
2. Comparison Operator
3. Boolean Operator
4. Bitwise Operator
5. Compound Operator

(1) Arithmetic Operator:

Operator name	Sign	A = 10, B = 20
Assignment	=	A = B [A = 20]
Addition	+	A + B = 30
subtraction	-	A - B = -10
Multiplication	*	A * B = 200
Division	/	B/A = 2
Modulo	%	B % A = 0

(2) Comparison Operation:

operator name	Sign	A = 10; B = 20 (A == B) is not true
equal	==	(A == B), is not true
not equal	!=	(A != B), is true
less than	<	(A < B), is true
greater than	>	(A > B), is not true
less than equal to	<=	(A <= B), is true
greater than or equal to	>=	(A >= B), false

③ Boolean Operation:

operator name	sign	A=10 ; B=20
and (logical)	&&	(A && B), is true (A: True, B: True)
OR (logical)	||	(A || B), is true
not	!	! (A && B), false ; ! true, false

④ Bitwise Operator:

operator name	Sign	A = 6 ; B = 8
Binary and	&	A & B = 0 (0110 = 6, 1000 = 8, 0000)
Binary OR	|	A | B = 1110 (0110 = 6, 1000 = 8, 1110)
Binary XOR	^	(A ^ B) = 1110
		Same হলে 0, ভিন্ন হলে 1
Binary not	~	~A = -6
		A = 6 [0 1 1 0]
		~A = -6 [1 0 0 1]
Right shift	>>	A >> 2
		A = [0 1 1 0]
		A >> 2 [0 0 0 1] = 1
left shift	<<	A << 2
		A = [0 1 1 0]
		A << 2 [0 0 0 1 1 0 0 0] = 24

⑤ Compound Operator:

B = 2
A = 5

operator name	sign	Example
increment	++	A++, 6
decrement	--	A--, 4
Compound addition	+=	B+=A, 7
Compound subtraction	-=	B-=A, -3
Compound multiplication	*=	B*=A, 10
Compound division	/=	A/=B, 2
Compound modulo	%=	A/=B, 1
Compound bitwise OR	|=	A|=B, 7
Compound bitwise And	&=	A&=B, 0

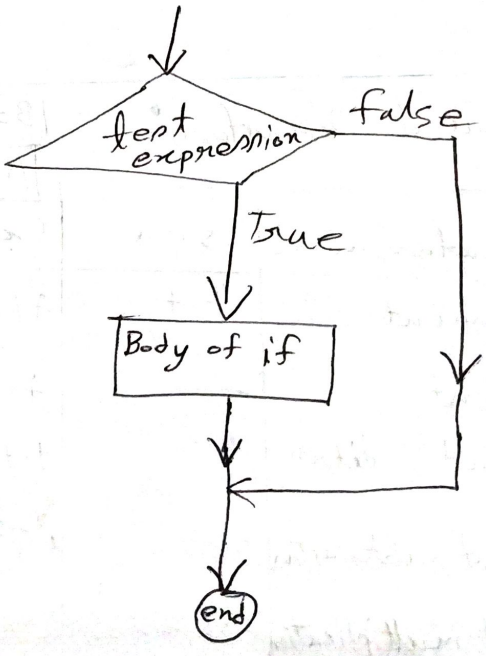
Class 4

Control statement:

- if
- if...else
- if...else...if()...else
- Switch case
- Conditional Operator (? :)

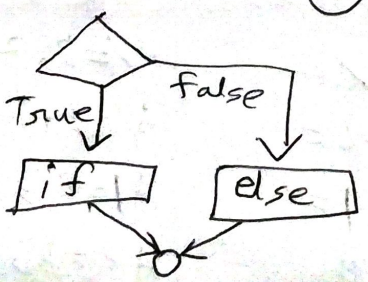
1. if statement:

```
if (expression/logic){
  Block of statements;
}
```



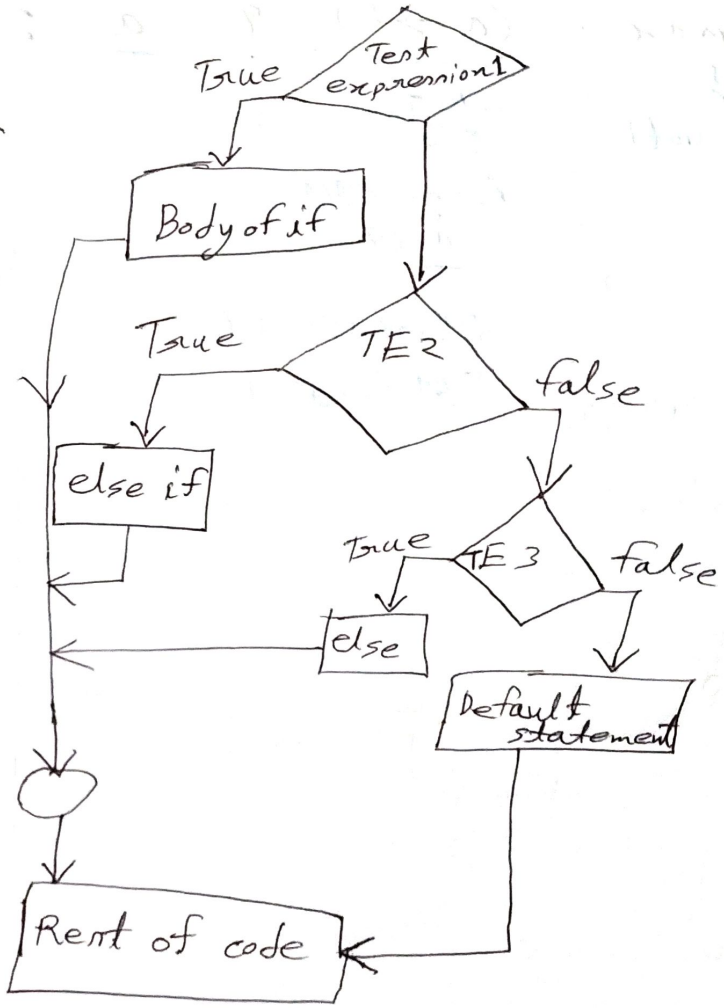
2. if....else statement:

```
if(expression){
   Block of statements;
}
else{
   Block of statements;}
```



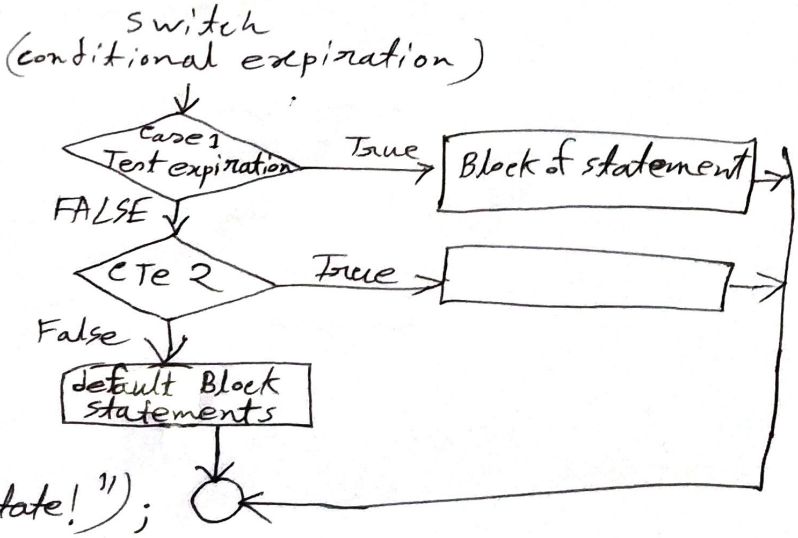
3. if....else...if.....else statement: (if else ladder)

```
if(expression_1){
   Block of statement
}
else if(expression 2){
   Block of statements
}
.
.
.
else{
   Block of statments;
}
```



Switch case statement:

```
switch(variable){
  case label:
       //statement
         break;
  case label:
       //statement
         break;
  default: Message("Invalid state!");
                 break;
}
```



Conditional operator (?:) : expression1 ? expression2 : ex3

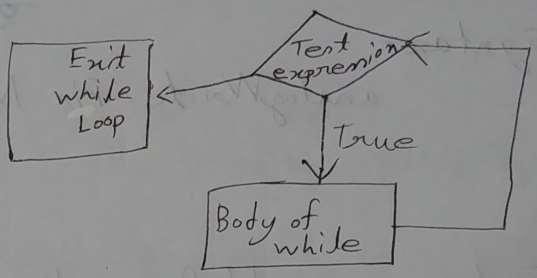
```
max = (a > b) ? a : b
```

- max → Variable
- (a > b) → যদি True হয়
- Assign করবে a কে
- নাহলে অর্থাৎ false হলে Assign b

Loops:

while loop:

```
while (expression){
  Block of statements;
}
```



Do...while:

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

for loop:

```
for(initialize; control ; increment or decrement){
   //statement block
}
```

Infinite loop:

```
for (;;){
  //statement
}
```

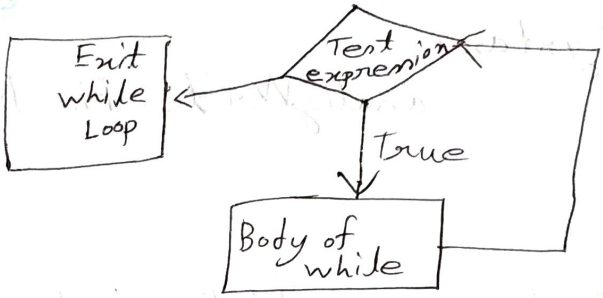
```
while(1){
  //statement
}
```

```
do{
  Block of statements;
}
while(1);
```

Loops:

while loop:

```
while (expression){
  Block of statements;
}
```



Do...while:

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

for loop:

```
for(initialize; control ; increment or decrement){
    //statement block
}
```

Infinite loop:

```
for (;;){
  //statement
}
```

```
while(1){
  //statement
}
```

```
do{
  Block of statements;
}
while(1);
```

Analog Write():

LED কে অন্ধ্র আলো থেকে Brightest আলো করা এবং বেশী থেকে অন্ধ্র করা হল Fading

Syntax-

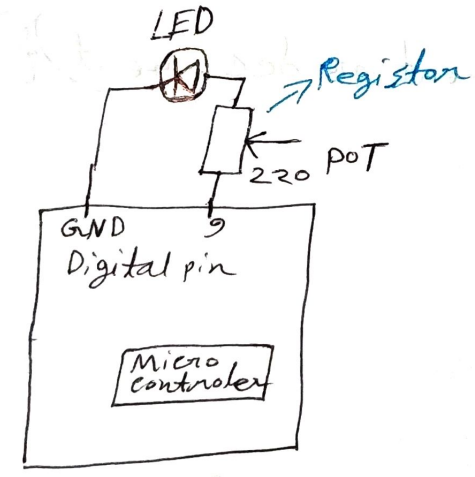
analogWrite(Pin Number, PWM value);

Fading a LED with analogWrite:

Hardware Required:

- Arduino or Genuino board
- LED
- 220 ohm resistor
- hook-up wires
- breadboard

circuit:



Arduino Uno

Code start:

```
int ledPin = 9;

void setup(){
  //nothing happens in setup
}
```

void loop

Now Arduino code for Fading : (analogWrite();)

```
void loop() {
for(int fadeValue=0; fadeValue<=255; fadeValue+=5){
   analogWrite (ledPin, fadeValue);
   delay (30);
  }
for(int fadeValue =255; fadeValue>=0; fadeValue-=5){
    analogWrite(ledPin, fadeValue);
    delay (30);
   }
}//end void loop.
```

class 5

What is Function?

Ans: Functions are the block of code which allows a programmer to create modular pieces of code that perform a defined task and then return to the area of code from which the function was called.

When necessary to create a function:

Ans:
The typical case for creating a function is when one needs to perform the same action multiple times in a program.

Advantages of function:

1. Functions help the programmer stay organized

2. Function codify one action in one place so that the function only has to be debugged once.

3. Reduces the chances for errors in modification.

4. Make the whole sketch smaller and more compact

5. Makes modular and more readable.

How to Write function:

```
Return_type function_name(argument1, argument2, .....)
{
//statements or function body
}
```

1 → Return_type
2 → function_name
3 → with type, argument name
4 → body

1. We can declare the function in two different ways —

How to prototyping a function:

```
Return_type function_name(argument type, argument type, ....);
```

1 → Return_type
2 → function_name
3 → int, double
4 → must be followed by a semicolon

Q. Write a sketch with function to find the average of two numbers:

Ans:

```
float Function(int x, int y){
   float result = (x+y)/2.0;
   return result;
}
void setup(){
  Serial.begin(9600);
}
Void loop(){
  int i=1;
  int j=2;

  float result = Function(i, j);
  Serial.println(result);
}
```

Serial Communication:

Use of Serial:

Used for communication between the Arduino board and a computer or other devices.

```
Serial.begin(9600); //opens serial port, sets data rate to 9600 bps
```

Serial.begin():

sets the data rate in bits per second for serial data transmission.

to communicate over pins 0 and 1 with a component that requires a particular baud rate.

***

After referring, Serial.begin(9600); we can communicate serially or we can say it opens the serial port. then we can use Serial.print("");

Serial.print( ) :

Serial.print(78); output : 78

Serial.print("Hello"); Hello

Serial.print('N'); N

Serial.print( ) can have 2 parameter.

code	output
Serial.print(78, BIN)	1001110
(78, OCT)	116
(78, DEC)	78
(78, HEX)	4E
(1.23456, 0)	1
(1.23456, 2)	1.23
(1.23456, 4)	1.2346

Example code: Printing decimal and binary of a number in tabular formate.

```
Void setup() {
   Serial.begin(9600);//open the serial port at 9600 bps
}
Void loop() {
 Serial.print("DEC");
 Serial.print("\t");
 Serial.print("BIN");
 Serial.print();//new line.

for(int x=0; x<10; x++) {
   Serial.print (x, DEC);
   Serial.print ("\t"); // prints a tab
   Serial.print(x, BIN);
   Serial.print( );//new line
   delay (200);
}
Serial.print( );
}
```

DEC BIN

Serial.write():

Writes binary data to the serial port. This data is sent as a byte or series of bytes

syntax:

```
Serial.write (val)
Serial.write (str)
Serial.write(buf, len)
```

code:

	output:
`Serial.write(72);`	H
`Serial.write(69);`	E
`Serial.write(76);`	L
`Serial.write(79);`	O

```
int bytesSent = Serial.write("Hellow");
//return the length of the string
```

Serial.println():

Print data to the serial port as human-readable ASCII text followed by a carriage return character and a newline character.

Syntax:

```
Serial.println(val)
Serial.println(val, format)
```

Code:

It is like Serial.print() but here a extra new line is printed.

Serial.available():

Get the number of bytes available for reading from the serial port. This is data that's already arrived and stored in the serial receive buffer.

Syntax:

```
Serial.available();
//reply only when you receive data
```

Serial.read():

Reads incoming serial data

Syntax:

```
Serial.read()
```

Code:

```
void setup() {
  Serial.begin(9600);
}
void setup(){
  if(Serial.available()>0){
    int incomingByte=Serial.read();}}
```

class 6

review class