Agenda-

```
Namespace
Cin and Cout
Functions
Function Overloading
Default Argumet
Class
DataMembers and Member Functions
Object
this Pointer
Acccess Specifier
```

Cin and Cout

```
cin is an external object of an istream class
cin use extraction operator(>>)
cin is a member of std namespace

cout is an external object of an ostream class
cout uses insertion operator(<<)
cout is a member of std namespace
```

Inline Function

```
These functions gets replaced by the compiler during call time.
It ensures faster execution.
It behaves similar to that of macros
Compile time also increases.
```

Function Overloading

```
functions having same name but different signatures
Rules for function overloading
    1. Function name should be same
    2. Different no of parameter
    3. Different type of parameters
    4. Differnt sequence of parameters
```

Default Argument Function

```
In this default values are given to the function parameters
Default values are given from right to left
```

Class

```
Building block that holds data and code together
class is a collection of datamembers and memberfunctions
in class by default everthing is private
class is a logical entity
class is a blueprint of an object
```

Object

```
1. State -> datamembers
    Values present inside this datamembers represents state of the object
2. Behaviour -> Member functions
    what operations we are able to do represents behaviour of object
3. Identity -> unique address
    If the state of objects is different/unique it can be considered for identity.
    If the state of objects are same then their actual address are considered for
identity.
```

Access Specifiers

```
1. private
2. public
3. protected -> we will see during inheritance
```

This Pointer

```
This is a constant pointer
It points to the current object.
Following functions dont get this pointer
    1. Global Function
    2. Static Function
    3. Friend Function
```

Memeber Functions

```
1. Constructor
2. Destructor
3. Mutators
4. Inspectors
5. Facilitators
```

Constructor

```
1. Default Ctor->Parameterless Ctor
    It is passes implicitely
    If user doesn't provide any ctor this Ctor is automatically passed.
a.Parameterless Ctor
    Ctor is called automatically during object creation
2.Parameterized Ctor

3. Copy Ctor
```