**Sunbeam Infotech**

**DAC : CPP - Day 1**

**\* History of C++**

**\* Object Oriented Programming Structure**

**\* Data types**

**\* Main Function**

**\* L Value and R-Value**

**\* Structure in C & in C++**

**\* Escape Sequence and Manipulators**

**\* Default Arguments**

**\* POP and OOP**

**\* Class and Object**

**\* Console Input and Output**

**\* History of C++**

-Inventor of C++ is Bjarne Stroustrup.

-C++ is derived from C and simula.

-Its initial name was "C With Classes".

-At is developed in "AT&T Bell Lab" in 1979.

-It is developed on Unix Operating System.

-In 1983 ANSI renamed "C With Classes" to C++.

**-C++ is objet orieted programming language**

**\* Few Real Time Applications**

Games

GUI Based Application (Adobe)

Database Software (MySQL Server)

OS (Apple OS)

Browser( Mozilla)

Google Applications(Google File System and Chrome browser)

Banking Applications

Compilers

Embded Systems(smart watches, MP3 players, GPS systems)

**\*SDK (Software Development Kit)**

- SDK = Language tools + Documentation + Supporting Library + Runtime Env.

int main() {}

int main(int argc, char \*\*argv , char\*\* env)

# Language Tools:

-Editor (to develop/edit source code)

-Preprocessor (To remove preprocessors-remove comments/expand macros)

-Compiler (Conversion of high level language into low level code( Assembly ))

-Assembler (Conversion of low level code into machine code)

-Linker

-Loader

-Debugger

# Documentation (MSDN / man pages)

**\* Object Oriented Programming Structure**

-OOPS is not a syntax. It is a process / programming methodology which is used to solve real world problems.

**-It is invented by Dr. Alan Kay. He is inventor of Simula too.**

**\* Data Types in C++**

It describes 3 things about variable / object

1. Memory : How much memory is required to store the data.

2. Nature : Which type of data memory can store

3. Operation : Which operations are allowed to perform on data stored inside memory.

- Fundamental Data Types (void, int,char,float,double)

-Derived Data Types ( Array, Function, Pointer, Union ,Structure)

- Two more additional data types that c++ supports are **bool , wchar\_t** (wide Character [one wide character : Size is 4 bytes)

**e.g: bool val=true;**

**wchar\_t**: Wide Character. This should be avoided because its size is implementation defined and not reliable.

Wide char is similar to char data type, except that wide char take up twice the space and can take on much larger values as a result. char can take 256 values which corresponds to entries in the ASCII table. On the other hand, wide char can take on 65536 values which corresponds to UNICODE values which is a recent international standard which allows for the encoding of characters for virtually all languages and commonly used symbols.

The type for character constants is char, the type for wide character is wchar\_t.

1. This data type occupies 2 or 4 bytes depending on the compiler being used.
2. Mostly the wchar\_t datatype is used when international languages like Japanese are used.
3. This data type occupies 2 or 4 bytes depending on the compiler being used.

* L is the prefix for wide character literals and wide-character string literals which tells the compiler that that the char or string is of type wide-char.
* w is prefixed in operations like scanning (**wcin)** or printing (**wcout)** while operating wide-char type.

Wide char type array or string:

**\* Main Function**

- main should be entry point function of C/C++

-Calling/invoking main function is responsibility of operating system. Hence it is also called as Callback function

**\* Comments**

**\* L-Value( Locator Value )**

Non constant( editable/modifiable) memory location which is available at left hand side of assignment operator is called locator value(L-Value).

Consider Following code:

5 + 3 = 8; //Error - L-Value Required

8 = 5 + 3; //Error - L-Value Required

const int number = 10;

number = number + 5; //Error - L-Value Required

**L-value is one of the following:**

1. The name of the variable of any type i.e, an identifier of integral, floating, pointer, structure, or union type.
2. A subscript ([ ]) expression that does not evaluate to an array.
3. A unary-indirection (\*) expression that does not refer to an array
4. An l-value expression in parentheses.
5. A const object (a nonmodifiable l-value).
6. The result of indirection through a pointer, provided that it isn’t a function pointer.
7. The result of member access through pointer(-> or .)

**\* R-Value( Reference Value)**

A constant, variable or expression which is used at right hand side of assignment operator is called R-Value.

r-value” refers to data value that is stored at some address in memory.

A r-value is an expression that can’t have a value assigned to it which means r-value can appear on right but not on left hand side of an assignment operator(=).

int num1 = 10; //10 - R-Value

int num2 = num1; //num1 – R-Value

int num3 = num1 + num2; //(num1 + num2) - R Value

**\* Structure in C & C++**

|  |  |
| --- | --- |
| **struct in c** | **struct in c ++** |
| we can include only variables into the structure. | we can include the variables as well as the functions in structure. |
| We need to pass a structure variable by value or by address to the functions. | We don't pass the structure variable to the functions to accept it / display it.The functions inside the struct are called with the variable and DOT operator. |
| By default all the variables of structure are accessible outside the structure. ( using structure variable name) | By default all the members are accessible outside the structure, but we can restrict their access by applying the keywords private /public/ protected. |
| struct Time t1; | struct Time t1; |
| AcceptTime(struct Time &t1); | t1.AcceptTime(); //function call |

**\* Escape Sequence and Manipulators**

Manipulatorsare helping functions that can modify the input/output stream. It does not mean that we change the value of a variable, it only modifies the I/O stream using insertion (<<) and extraction (>>) operators.

Header File : #include<iomanip> // input output manipulation

Setbase(16) or hex

setbase(8) or oct

setbase(10) or dec

endl

\b , \t , \n , \\, \’ , \” ( Escape sequences)

setw (val)

setfill(char c)

setprecision (val)

**\* Default Arguments**

- A default argument is a default value provided for a function parameter/argument.

- If the user does not supply an explicit argument for a parameter with a default argument, the default value will be used.

- If the user does supply an argument for the parameter, the user-supplied argument is used.

**\* POP and OOP**

|  |  |
| --- | --- |
| **OOP** | **POP** |
| OOP stands for Object Oriented Programing. | POP stands for Procedural Oriented Programming. |
| OOP follows bottom up approach. | OOP follows top down approach. |
| A program is divided to objects and their interactions. | A program is divided into funtions and they interacts. |
| Objects communicates with each other by passing messeges. | Functions communicate with each other by passing parameters. |
| Inheritance is supported. | Inheritance is not supported. |
| Access control is supported via access modifiers.  (private/ public/ protected) | No access modifiers are supported. |
| Encapsulation is used to hide data. | No data hiding present. Data is globally accessible. |
| C++, Java | C , Pascal |
| It overloads functions, constructors, and operators. | Neither it overload functions nor operators |
| Classes or function can become a friend of another class with the keyword "friend". Note: "**friend**" keyword is used only in c++ | No concept of friend function. |
| Concept of virtual function appear during inheritance. | No concept of virtual classes . |

**\* Class and Object**

**Class:**

-Class is collection of data member and member function.

-Class represents set/group of such objects which is having common structure and common

behavior.

-class is logical entity.

- Class contain (Nested Type[enum,class,structure,union],Data Members,Member Function]

**Object :**

- Instance of class is object

-An entity, which get space inside memory is called object.

-Object is used to access data members and member function of the class

-Process of creating object from a class is called instantiation

**- Object has**

1. State

- Value stored inside object is called state of the object.

- Value of data member represent state of the object.

2. Behavior

- Set of operation that we perform on object is called behavior of an

object.

- Member function of class represent behavior of the object.

3. Identity

- Value of any data member, which is used to identify object uniquly

is called its identity.

- If state of object is same the its address can be considered as its

identity.

-Member function do not get space inside object.

-If we create object of the class then only data members get space inside object. Hence size of object is depends on size of all the data members declared inside class.

-Data members get space once per object according to the order of data member declaration.

-Structure of the object is depends on data members declared inside class.

-Member function do not get space per object rather it gets space on code segment and all the objects of same class share single copy of it.

-Member function's of the class defines behavior of the object.

**\* Data Members**

-Data member is also called as field, attribute, property etc.

**\*Member Functions**

-A function implemented inside class scope is called member function

-Member function is also called as method, operation, behavior or message.