

# The Best Way to Teach C++

**Er.Yadwinder Kaur**

**Er.Satvir Singh**

**Abstract:** It has been analyzed that C++ is a dreaded object for students forever. Most of the students cannot find interest in C++. There may be many reasons of students' disinterest in C++ –Teacher does not relate the programming basic constructs with things used in real life time, students' mentality to stay away from programming as a cup of tea is not meant for everyone etc. Researchers have understood this difficulty and proposed various algorithms and tools to overcome it [1-2]. Programming is mathematical by nature. To solve complex real world problems, algorithms are designed. Interactive technology has facilitated relationship building between real world and programming [3]. The present paper will help teachers to use a basic method while teaching programming in C++. Proposed paper gives a few examples that a human being encounters in daily life. This paper includes the reasons behind not learning these languages.

**Keywords:** Programming Languages, C++, Classes, Objects, Programming's Basic Constructs.

## I. INTRODUCTION TO PROGRAMMING LANGUAGES

A programming language is the way to communicate with computer hardware. Most of the languages require translators, as computer can understand only machine language. Compilers, interpreters, assemblers, linkers and loaders are used as translators.

There are three languages, if human being is considered in communication:

High Level Languages (HLL): The languages which are only be understood by human being. For example- Hindi, Punjabi, English, French etc.

Middle Level Languages (MLL): The languages which are neither understood by human being nor by machine. For example – C, C++, Java etc.

Low Level Language (LLL): The Languages which are understood by machine only. For example- Machine Language.

There are two languages, if human being is not considered in communication:

High Level Languages (HLL): The languages which are not understood by machine. For example- C, C++, Java, Cobol etc.

Low Level Language (LLL): The Languages which are understood by machine only. For example- Machine Language.

## Use of Programming Languages

Programming languages are the properties which show what required to build programming languages and used to create programs. Programs are the collection of instructions. Programs are used to control the machine's execution process and to execute algorithms more precisely.

## II. COMMON METHOD AND FEW REASONS OF FAILURE TO TEACH AND LEARN LANGUAGES

It has been come into our attention that we all are following the common outline to teach these languages.

### For C:

- 1) Fundamentals
- 2) Control structures
- 3) Functions
- 4) Arrays and Strings
- 5) Structures and Unions
- 6) Pointers
- 7) File handling

### For C++:

- 1) Fundamentals
- 2) Control structures
- 3) Functions
- 4) Arrays and Strings
- 5) Structures and Unions
- 6) Pointers
- 7) Objects and Classes

- 8) Constructors and Destructors
- 9) Operator Overloading and Type Conversion
- 10) Inheritance
- 11) Virtual functions
- 12) File handling
- 13) Exception handling
- 14) Templates

Authors have observed that teachers are not teaching these subjects effectively. Our education's syllabi and pressure on teachers to complete and move as per syllabi has played biggest role to create this situation. In India most of the books are published only as per universities' syllabus. Students inclinations are towards to only clear the subjects in exams rather to learn them. "Engineers do fight at last night" is the famous slogan among young people pursuing Bachelors or Masters in engineering. Before the commencement of exams students buy the book contains only syllabus related topics, rote them, attempts the exam and fetched the CGPA either equal or more than 8 or 9 easily. Now here we are also failing in setting the exams questioner.

### III. PROPOSED METHOD TO TEACH

To reduce the complexity and to enhance the interest in subject, authors have proposed some examples, might referred by teachers while delivering lectures. All example encounters in daily life.

Example for 'for loop':

"You are a first year student of B.Tech, Computer science and engineering, of 2014 Batch. Every year you have to study 10 subjects. Until you end your graduation, i.e. 2018, you will have studied 40 subjects. It means study of 10 subjects is the repeated task for 4 years and after 4 years this process is ended."

In the figure person sunny is sleeping now. it is 7 o'clock on the clock. At 7:30 AM sunny wakes up. In advance we know till what time sunny is going to wake up. So in this case for loop is used. But if in advance time of process continuation is not known before time, while or do-while is used.



"while it's raining, do use your umbrella". It can't be predicated in advance at what time rain ends. In this case while loop must be used, as work totally depend upon the condition. while loop totally conditioned based.



### If else

Raghav wants to attend a wedding ceremony and he is already late. Venue is 40KM away from his house. He thinks: "If I had a car, I would have reached there early". Now there are two solutions to the problem "reached": Reached early or reached late. If he got a car he reaches early, else he reaches late.

### Switch case

A person, Amar, is in Mumbai. He wants to travel to Delhi. There are three modes of journey from Mumbai to Delhi. Amar has three choices: By car, by plane and by train. Now decision has to be made. if he opts Mode I, he will travel by car or if he opts Mode II, he will travel by plane or if he opts Mode III, he will travel by train. Switch is used when choice is made among many options.



Mode – I

**Choice 1:** If Amar opts for first choice, he will travel by a car



**Mode – II**

**Choice 2:** If Amar opts for second choice, he will travel by a plane



**Mode – III**

**Choice 3:** If Amar opts for third choice, he will travel by a train

Take another example of switch board in a room with three switches. First switch is connected with fan, second switch with light and third is with exhaust fan. Now user of the room has been given three choices: Fan, light and exhaust fan. If switch one is pressed, fan will be on. If switch two is pressed, light will be on and if third switch is on, exhaust fan will start working.

### Functions

Function is the toughest topic to understand for students in every programming. Function are known with different names in different Programming Languages. Most of languages used them as methods, procedures, modules, subprograms etc.

To explain functions an example of cooking may be taken. A Chef Raman wants to cook “*IdliSamber*” within 45 minutes. To speed up the task chef decides to break up the task in his team

members. He hands over the chopping task of vegetables to Ajay, chopping and grinding of spices to Ritu and selects Neeru to cook Idli. He choose himself for boiling the Pulse, Chaunk to Sambhar and preparation of Idlimixture.

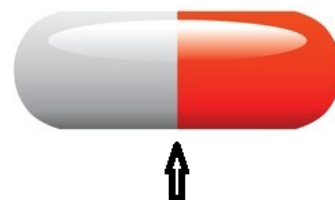
Now every member has his/her dedicated task (task can be compare with function in C++) and everyone will work independently but finally will be joined by Chef Raman (Raman works as main function). While supervising the task Raman asks the members to start the given task at right time (evoke a function). To get started a task he may provide ingredients (passing argument) and after completion of the task he may ask the member to hand over the ready material to him (data returned by function).

### Classes and Objects

Best way to teach the above cited topics; take the example of two student classes: Class I and Class II. The class is the collection of objects and object is an instance of a class. Everything which is having some mass and acquires some space is called an object. So in figures: students, teachers, boards, newspaper, benches etc. are the objects and collectively make a class.

Both classes are **abstract** class for each other unless any of them has have student/teacher object of other class. Both classes have an idea that another class is being going on. But unaware about number of students seating and topic being taught in another class.

**Encapsulation's** can be explained by taking the example of capsule. Acap keeps medicine and its operation, means remedy for disease, together and cap is named as capsule.



*Medicine + its operation together*

Objects are independent from each other. Teacher object delivers his/her lecture and student objects' have their independent grasping power and will to write notes. If one student pen down anything into

his/her notebook, it is not written down anyone else's notebook. one object doesn't have any interference on other object.



**Class-I**



**Class-II**

#### IV. CONCLUSION

It's been tedious job to learn Programming language for students. Both teachers and students encounter difficulties to teach and learn programming respectively. Teaching programming can be made interesting by introducing real world examples in lectures. Paper has excluded general syntaxes of different programming constructs. Some authors have used games in other systems for teaching programming, in [4] word games have been used that manipulate strings and number and games that manipulate integers and random numbers. Some animated drawing and video can be the part of lectures. The Main motive of the proposed method to bring the interest of students into the subject.

#### REFERENCES

- [1] Bloom, B., Englehart, M. Furst, E., Hill, W., & Krathwohl, D. (1956). Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain. New York, Toronto: Longmans, Green.
- [2] Kelleher, C., Pausch, R., (2005). Lowering the barriers to programming: A Taxonomy of programming environments and languages for novice programmers, ACM Computing Surveys, Vol.37, No. 2, p.83-137.
- [3] Duffy, T. M., & Cunningham, D. J. (1996). Constructivism: Implications for the design and delivery of instruction. In D. H. Jonassen (Ed.), Handbook of research for educational communications and technology. New York: Macmillan.
- [4] Rajarivarma, R. (2005). A games-based approach for teaching the introductory programming course, ACM SIGCSE Bulletin, Vol.37, No.4, p.98-102