Programming in C++

SWAYAM Prabha Course Code: R13

PROFESSOR'S NAME	Dr. Partha Pratim Das
DEPARTMENT	Computer Science and Engineering
INSTITUTE	IIT Kharagpur

COURSE OUTLINE

There has been a continual debate on which programming language/s to learn, to use. As the latest TIOBE Index for April 2016 indicates – Java (21%), C (14%), C++ (6%), C#(4%), and Python (3%) together control nearly half the programming community. Given this, it is still important to learn C and C++ because of the efficiency they offer. While we appreciate that Java is good for applications, for graphics; and we acknowledge that Python is appropriate for portable software, engineering problem solving, and graphics; it is worth bearing in mind that the JVM and Python interpreter are indeed written in C++, making C++ the father of all languages today. Well, hence, C++ is the systems language. Why should I learn it if my primary focus is on applications? This is where the recent updates of C++, namely, C++11, C++14, and C++17 offer excellent depths and flexibility for C++ that no language can match. These extensions attempt to alleviate some of the long-standing shortcomings for C++ including porous resource management, error-prone pointer handling, expression semantics and better readability. The present course builds up on the knowledge of C programming and basic data structure (array, list, stack, queue etc.) to create a strong familiarity with C++98 and C++03. Besides the constructs, syntax and semantics of C++ (over C), we also focus on various idioms of C++ and attempt to go to depth with every C++ feature justifying and illustrating them with several examples and assignment problems. On the way, we illustrate various OOP concepts.

While this course can be understood independently (after a course in C programming), it would help in developing understanding in OOP. Hence this course is advised in conjunction with OOP.

Course Outline

1. Programming in C++ is Fun: Build and execute a C program in C++, Write equivalent programs in C++.
2. C++ as Better C: Procedural Extensions of C.
3. Overview of OOP in C++: Classes and basic Object-

	Oriented features (encapsulation). 4. Overview of OOP in C++: More OO features, overloading, namespace and using struct and union. 5. Inheritance: Generalization / Specialization of Object Modeling in C++. 6. Polymorphism: Static and Dynamic Binding. 7. Type Casting & Exceptions: C++ cast operators; C++ Exceptions & standard exception classes. 8. Templates & STL – Function and Class templates and using STL like containers, algorithms.