

APT 1030: FUNDAMENTALS OF PROGRAMMING LANGUAGES

Prerequisite: IST 1020

1 credit Unit

Course Rationale

Students need a firm grasp of the basics of computer programming before attempting to learn a specific language. This course is intended for students who have no previous programming experience to understand the basics of computer programming. The concepts, techniques and syntax taught in the course are applicable to all computer languages, and will assist in student success in other, more advanced programming classes such as Visual BASIC, JavaScript and C++. The aim of the course is to expose the students to the general aspects of programming languages.

Course Description

This course acts as a guide to understanding computer languages design in general. This course presents concepts and structures governing the design and implementation of modern programming languages, run-time representations of traditional block structured languages, typing systems, abstraction and procedure mechanisms, and storage management. It also introduces language design issues and language translators. Course also provides fundamental introduction to computer programming theory and concepts to students with little or no previous experience. Students learn structure, syntax, logic, and the difference between object-oriented and procedural systems methodologies. Students are introduced to structured programming OOP as well as machine organization. A high level language such Visual Basic is suggested as introductory language but other languages such as C/C++ may be used with justification.

Course learning outcomes

At the end of the course, students will be able to:

1. Identify structures governing the design and implementation of modern programming languages
2. Describe how different programs are compiled and executed
3. Differentiate the features of 1st, 2nd, 3rd, 4th, 5th and 6th generation programming languages.
4. Explain the motivation for continuing evolution of programming languages with modern examples including move from procedural to object-oriented, to component-based, and to web services.
5. Design and use simple algorithms using flow charts and pseudocodes.
6. Describe the features and rationale of Object Oriented Programming.

7. Use a popular programming language such as Java, C# or Visual Basic to implement simple computer programmes.
8. Use a modern programming language, demonstrate competency in basic programming skills including use of variables, constants, control structures, simple data types, input and output management, and simple GUIs.
9. Apply the standard computer programme development cycle of specification, design, implementation, testing/debugging, and maintenance using a modern programming language.
10. Demonstrate ability to appropriately document programming code and conform to recommended coding style/conventions.

Course Content

Introduction to programming: Program structure, variables, simple functions. Writing, compiling and executing programs; Expressions, simple control structures: decision making, looping and flow control, input and output. Simple algorithm development. Arrays; More on control structures; Introduction to OOP concepts and graphical user interfaces. Practical skills in using contemporary software development environments. Debugging, compiling, and software distribution. Writing simple programs to solve simple practical problems, Concepts and structures governing the design and implementation of modern programming languages, Language design issues and language translators, Programs compilations, Program executions

Teaching Methodologies

Lectures, Presentations by members of the class, Case discussions, Tutorials, Assignments, Continuous assessment tests, Lab Practical, Library, appropriate software, manual/notes, simple projects.

Instructional Materials/Equipment

Course text, Handouts, White board, Presentation slides, Journals

Methods of evaluation

Class assignments, take-home assignments, tests, small projects to demonstrate use of software tools.

Laboratory Work	20%
Project	20%
Assignments	10%
Mid-semester	20%
Final semester exams	30%

Total

100%

Course Text

Fundamentals of programming Languages by **Dipali P. Baviskar 1st Edition 2009**

Recommended reading

Programming language Fundamentals by example by **D.E Stevenson 2006**

Fundamentals of Programming Using Java by **Edward Currie 2006**

H.M. Deitel and P.J. Deitel **Visual Basic 2005 How to Program**, (New Jersey, Prentice-Hall, 2006)

Sprankle, Maureen. **Problem Solving and Programming Concepts**. 6th ed. New Jersey: Prentice Hall, 2003.

Internet resources.

Kenneth C. Loudon (1993) Programming Languages: Principles and Practice, PWS-Publishing Company; 2nd Edition (July 15, 2002) ISBN: 0534953417