

e-tunity Courseware

e-tunity

2001 ff.

The following overview is a list of our courses. Per course, a general overview is given. For more information, such as the duration of the course and for a more detailed description, please don't hesitate to contact us via info@e-tunity.com.

C 01: *C Basics*

Duration

3-5 days, depending on preexisting knowledge of students

Prerequisites

For this course, no knowledge of C whatsoever is required. Some knowledge of programming languages (e.g., VB, Perl or the like) is helpful.

Contents

- Platforms and Software
- Documentation: K&R, man, info, ...
- Tools of the trade: the preprocessor, compiler, linker, debugger
- The C syntax
 - Basic types, Variables, statements and expressions
 - Functions, main()
- Using the C libraries: includes and libs
- Functional decomposition and modular development
- Operating system interface
- Introduction to pointers

General information

In this course, students learn the basics of the C programming language. They are able to use all related tooling, and to write and maintain simple C programs, and are able to find their way in existing documentation on C functions and libraries.

C 02: Advanced C

Duration

5 days

Prerequisites

Course C01 or comparable knowledge.

Contents

- In-depth pointers
 - call-by-reference, pointers to pointers
 - Arrays of pointers, pointers to arrays
 - Functions returning pointers
- Memory management: allocation and de-allocation, memory leaks²
- Recursivity
- Complex data structures
 - Lists
 - Trees
 - Hashes
- Automatic program construction: make

General information

This course follows C01. After completing this course, students have a thorough comprehension of pointers and complex data structures, and are able to define and create specific data structure for specific purposes. Some of the advanced programming techniques are highlighted. The course is intensive and requires an affinity with programming.

C++ 01: Basic C++

Duration

10 days

Prerequisites

For this course, a thorough knowledge of the **C** programming language is required. The students must have a good understanding of pointers, pointers to functions, functions returning pointers, etc.. Furthermore the knowledge of the memory allocation model of **C** is required (this is the *malloc()* and *free()* family of functions).

Contents

- Differences with plain C
- Classes and members: public, private, friends

-
- Operator overloading
 - Constructors and Destructors
 - Memory management in C++
 - C++-style IO: Streams

General information

In this course, students learn to maintain and write C++ programs. The following concepts of C++ are discussed in detail: data encapsulation and data hiding, memory allocation in C++, including the overloading of the assignment operator and the copy-constructor function, and file IO in C++ (streams). This course does not focus on specific implementations of C++ class libraries, e.g., GUI-building. Instead, the students learn how to approach such tools and how to use them.

Java 01: *Java Basics*

Duration

2 days

Prerequisites

Some programming experience with a 3GL or 4GL programming language or environment (e.g. VB, Fortran, Pascal, C, Access).

Contents

- Your first cup of Java
- Object Orientation
- Java tools and Utilities
- Introduction to the Java environment and libraries

General information

After the completion of the course, students know the basics of Java, know how to find the way through the Java documentation and have a general understanding of object orientation and the Java environment.

Java 02: *Advanced Java*

Duration

3 days

Prerequisites

Some experience with writing Java programs (not Javascript!). Knowledge of the general syntax of Java, basics of Object Orientation.

Contents

-
- Inheritance revisited
 - Constructor details
 - Inner classes
 - Exceptions and error handling
 - Packages
 - Modifiers and Visibility
 - I/O
 - Collections
 - Using Javadoc
 - Programming standards
 - Introduction to Javabeans
 - Introduction to Servlets
 - Introduction to SWING

The contents consists of separate modules and can be partly adapted to the specific needs and wishes of the students.

General information

In this course students learn how to use the advanced features of the Java language and the more general parts of the Java platform. Some of the more important libraries and utilities are exploited in depth.

Gram 01: *Introduction to Grammars*

Duration

3 days

Prerequisites

C 02 or comparable expert knowledge of the C programming language

Contents

- Formal grammar rules: LALR-1 grammars
- Definition basics: repetition, alteration, selection, optionality
- Associativity and Precedence
- Tools of the trade
 - yacc (bison)
 - lex (flex)
- Expression analysis

General information

After completing this course, the students can design and create parsers of formal languages. During the course, a parser for one formal language is designed and coded. The students work on a Linux system using the "standard" tools *flex* and *bison*. Optionally, in-depth information will be given on expression crunching and on bytecode generation using a stack-machine.

Perl 01: *Perl Basics*

Duration

3 days

Prerequisites

General knowledge of operating systems (preferably Unix), and knowledge of one or more programming languages.

Contents

- Platforms and Software
- Documentation: man, perldoc
- Structure of Perl programs
 - Statements, variables and functions
 - Types, variables and operators
 - Flow control and subroutines
- File IO and operating system interface
- Strings and Regular expressions
- Using basic Perl modules
- (Optional) Installing new Perl modules

General information

After the completion of the course, students can write or maintain simple Perl programs. Using Perl modules is also discussed. This course is a very good addition to the Unix courses, as Perl is one of the most widely used scripting languages on Unix.

Perl 02: *Advanced Perl*

Duration

3 days

Prerequisites

Perl 01 or comparable knowledge

Contents

- Writing basic Perl modules
- Embedded documentation
- The Perl Object Model
 - Classes, objects, methods and instances
 - Using OO-modules
- Writing OO-modules

-
- Instance-bound variables and functions
 - set/get functions
 - Constructors and destructors
 - Class-bound variables
 - Extending classes: inheritance

General information

This course is an extension of Perl 01, and focuses on the more "advanced" features of Perl. Once Perl programs grow larger, the need for an object-oriented approach also grows. Using the Perl OO Model, the students learn how to define Perl classes, instantiate variables, use data encapsulation.

Perl 03: *Perl for the Internet*

Contents

- HTML: documents vs. forms
- Form elements and submitting
- Posting methods
- The CGI standard
- Retrieving posted variables
- Common processing tasks
 - Outputting HTML
 - Parsing existing pages
 - Form generation from Perl: CGI.pm
- (Optional) Basic user management for sites

PHP 01: *PHP Basics*

Duration

3 days

Prerequisites

General knowledge of IT, SQL01 or comparable knowledge is recommended

Contents

- Platforms and Software
- Documentation
- Structure of PHP programs
 - Types, variables and constants
 - Operators
 - Flow control

-
- Functions and subroutines
 - Classes and Objects
 - Predefined functions
 - Strings and Regular expressions
 - File I/O and operating system interface
 - Using a database

General information

After the completion of the course, students can write or maintain simple PHP programs. They can use file I/O functions and manipulate data in a MySQL database.

PM 01: *Project Management using Prince 2*

Duration

2 days

Prerequisites

General knowledge of project management of development projects, either as a project manager, or as a project engineer.

Contents

- What is Prince 2
- What is **not** Prince 2
- The Prince 2 Components
 - Organization
 - Plans
 - Stages
 - Controls
 - Risk management
 - Quality issues
 - Configuration management
 - Change control
- The Prince 2 Processes
 - Starting up a project
 - Planning
 - Directing a project
 - Controlling a stage
 - Managing product delivery
 - Managing stage boundaries
 - Closing a project
- Techniques
 - Product-based planning

-
- Project filing techniques
 - Quality review techniques
 - Change control approach

General information

In this course the students learn the necessity of a structured project management methodology when developing software. Prince 2 is presented in the course as an example of how a well organized project may look like. Since the position of a Prince 2 structured project within an organization is very well defined, attention is paid to how a project can 'survive' in a dynamic and thus changing organization.

After completion the students understand why a structured project management methodology like Prince 2 is required to setup and manage a project. The students understand the existence and definitions of Prince 2 Components and Processes and are able to apply those on projects in their own environment.

SQL 01: *SQL Basics*

Duration

2 days

Prerequisites

General knowledge of IT

Contents

- Basic Principles of SQL - what is SQL and what can you do with it.
- Databases
 - Databases
 - Datatypes
 - Table, columns and rows
 - Keys
- Selecting data
 - Selecting
 - Filtering
 - Sorting
- Manipulating data
 - Insert
 - Update
 - Delete
- Data definition language (DDL)
- Indices
- Functions

General information

After the completion of the course, students can write or maintain simple SQL programs. The course information is not database-vendor specific; the students can apply their knowledge to several databases.

Unix 01: *Unix Basics***Duration**

2 days

Prerequisites

General knowledge of computers and an operating system, preferably having a command-line based interface, such as MS-DOS, VAX/VMS, Unix

Contents

- The login procedure: username, password
- Logins: console, telnet, ssh
- The login shell
- Basic commands: cd, ls, cp, mv, rm
- Getting help: man, info
- The filesystem layout: /tmp, binaries, homes
- User management: useradd, userdel, passwd
- Archiving: tar, ftp, scp
- (Optional) The working environment: profiles, prompt, aliases

General information

After the completion of this course, students can 'find their way' around a Unix system, and they can perform simple system administration tasks such as user management, making backups and restoring them. The course is given on Linux systems where each student is given their own working area to exercise.

Unix 02: *Unix Expert***Duration**

3 days

Prerequisites

Course Unix 01, or comparable knowledge

Contents

- The working environment: profiles, prompt, aliases

-
- E-mail readers: mail, pine, elm
 - E-mail configuration: forwarding, sendmail configuration
 - Logs: /var/log
 - Monitoring logs: grep c.s.
 - Job scheduling: cron, at
 - Triggers: disk full, etc.
 - Cleaning up the system (/tmp, corefiles, logs)

General information

After the completion of the course, the students can perform complex system administration on a Unix system, such as e-mail configuration, log checking and administration of scheduled tasks. Furthermore the student learns how to build 'triggers' that perform a given action when a condition (such as a high disk usage) on the Unix system occurs.

Unix 03: *Unix Scripting*

Duration

3 days

Prerequisites

Unix 02 or comparable knowledge

Contents

- The Unix filter model: everything's a file
- Redirection and piping
- Simple filters: grep, tr, recode
- The shell script language
 - Program arguments and variables
 - Functions
 - Flow constructs: if, while, case
 - Testing for return values
- More complex manipulation: awk
 - Fields and records
 - Functions

General information

This course focuses on the file-oriented approach to Unix and on data streams. The appropriate scripting files are discussed. This course is a good "second-step" to Unix 01. Combining the knowledge of both courses, the students are able to perform complex administrative tasks.

Unix 04: *Unix Security*

Duration

2 days

Prerequisites

Unix 02 or comparable knowledge

Contents

- General security aspects: passwords, users
- Security triggers
- Network security
 - Services: ports and servers, most common ones
 - What is a firewall
 - A two-point firewall: filtering and masquerading
 - A three-point firewall: DMZ
- (Optional) E-mail security and anti-spam measures

General information

This course focuses on the security aspects of Unix systems. After completion, the students have insight in the general security aspects of a Unix system, such as a secure user and password scheme. Furthermore aspects of a network-connected server are discussed, focusing on a two- or three-way firewall. The course is hosted on Linux servers (SuSE 7.2 or better), where each student has a root-account to exercise. Optionally, e-mail security and anti-spam measures are discussed.

XML 01: *Introduction to XML*

Duration

2 days

Prerequisites

General knowledge of HTML and CSS. Knowledge of the internals of Internet applications (e.g., of the CGI model) is an advantage.

Contents

- The purpose of XML
- Tags, data and attributes
- Designing an XML standard: the schema
- Transformations to HTML
 - Template matching
 - Iteration

-
- Selection
 - Sorting
 - Filtering and conditional transformations

General information

This course focuses on the translation of XML data using XSL and on displaying XML data in webpages. After completion of this course, students can design an XML schema and define matching XSL style sheets for display purposes.