Introduction to Practical Exercises

Overview

The practical exercises are designed to supplement the technical material in this course. Many of the questions are deliberately vague to encourage you to apply some 'artistic licence' in both the creation of tighter specifications and in the design of the software solution; there are no unique correct answers.

Each section describes overall objectives, reference material, an overview of the exercises and the questions themselves, including an optional section at the end.

Software Set Up

Windows

The exercises are found in the directory called **qacprg**. Each practical exercise is located in a suitably named sub-directory. For example:

The *Looping* exercise is located in: **c:\qacprg\looping**

The *Decisions* exercise is located in: **c:\qacprg\decision** and so on.

We will be using Visual Studio during the exercises. The environment allows you to write and compile C source code (as well as C++ source code). Visual Studio provides an editor where you can edit your code, plus an integrated compiler and linker to build and run your programs. Visual Studio organises your code into projects. We will provide a separate project for every question.

Note: if this course is not running on QA premises then you might be using a different version of Visual Studio, in which case your instructor will point out the differences.

At the end of the course, we will show you how to use our takeaway service, to copy all your code

Linux

A Linux VM is provided which runs under VMWare Player. It is based on CentOS (a Red Hat variant), and the folder name is "C_CPP CentOS *n.m*", where *n* and *m* are version numbers.

The use of Linux is optional, but assumes a basic knowledge of using the command-line (Bash).

Linux username should be **QA User** for everyone and the password **qa**, root's password is secret. In QA User's home directory is a directory for each course, using the same names as on Windows, in this case use directory **qacprg**.

For example:

The *Looping* exercise is located in: **/home/user1/qacprg/LOOPING**

The *Decisions* exercise is located in: **/home/user1/qacprg/DECISION**

and so on.

In nearly all the questions within each exercise, the instructions in the Practical Outline will start with, for example, *… on Windows open **numbers.sln**, on Linux edit **numbers.c** ....*

Note: if this course is not running on QA premises then your setup might not be as described here, in which case the instructor will describe the differences.

Each directory and lab sub-directory has a **Makefile**, which is a form of build script invoked by typing **make**: that way you don't have to call the compiler directly yourself (although you can if you want to).

In the top level directory is a script called **takecode**, which can be called passing the lab directory name as its first parameter. It will zip relevant source files for delegates to take away. Access to Windows C: drive is through a shared folder, **/mnt/hgfs/C**.

Eclipse for C/C++ is also installed, but no projects are built. Feel free to use it if you already have experience of Eclipse.

Help and Hints

Some of the questions invite you write a complete C program from scratch, but in some cases we provide a simple code template to get you going. These code templates relieve you of some of the mundane coding, so that you can concentrate on the relevant (and more interesting!) aspects of the program.

If there is any doubt, please ask your instructor.

Solutions

All the questions have sample solutions. Each working directory has an accompanying subdirectory called **solution**, which contains the solutions for each question. Feel free to look at these solutions as a further form of help.

Note also that as the exercises progress, assumptions are made that you are familiar with your environment, so there might well be fewer hints.

Paper and Pencil Exercises

Practical exercises on **Operators and Types**, **Pointers** and **Further Data Types** have 'paper and pencil' questions. The answers are all found on-line. It will be necessary to build and run these programs, to display the correct answers on the screen. Alternatively, of course, you can just ask your instructor!