Chris Heunen, 07/03/2022:

I'd like to propose updating the course descriptor for CQI (http://www.drps.ed.ac.uk/22-23/dpt/cxinfr11128.htm) for next year as attached.

Summary of changes: more emphasis (10%->30%) on use of actual quantum programming languages rather than just theory (90%->70%).

This is reflected in a new title ("Categories and Quantum Informatics" -> "Introduction to Quantum Programming and Semantics"), a new learning outcome, an updated assessment structure (25% lab work replacing current 25% coursework), and updated prerequisites (INF1b or equivalent).

Title:

Introduction to Quantum Programming and Semantics

Summary:

There are several languages for programming quantum protocols. Each has its own strengths and weaknesses. This course surveys current platforms (OpenQAsm, Qiskit, Q#, Quipper, Quantomatic, and PyZX) and analyses their respective features semantically. The theoretical analysis uses category theory, a powerful mathematical tool in logic and informatics, that has influenced the design of many modern programming languages. It enables a powerful graphical calculus that lets us draw pictures instead of writing algebraic expressions. This technique is visually extremely insightful, yet completely rigorous. For example, correctness of protocols often comes down to whether a picture is connected or disconnected, whether there is information flow from one end to another. In a practical way, this course investigates the conceptual reasons why quantum protocols and quantum computing work, rather than their algorithmic and complexity-theoretic aspects.

Course description:

The course has two parallel tracks: a practical one, and a theoretical one. The practical track surveys different quantum programming platforms. These include OpenQAsm, Qiskit, Q#, Quipper, Quantomatic, and PyZX. We discuss their basic structure, strengths, and weaknesses. Via labs and live coding in lectures, students get hands-on experience of implementing small quantum programs. The emphasis is on the primitive programming constructs and structure of each language, not on large-scale quantum software development.

Simultaneously, the theoretical track analyses the features of each language in denotational semantics, focusing on monoidal categories. Via lectures and self-study reading, the course teaches students the basics of dual objects in monoidal categories. Specific attention is paid to the graphical calculus, which makes the topic visually apparent. Via weekly exercise sheets, and their review incorporated into the contact hours, the student learns to graphically manipulate algebraic objects such as monoids and Frobenius structures. They will understand that this still allows perfectly rigorous proofs of correctness, and be able to see the information flow of a protocol that is often hidden behind superfluous details.

Throughout the course, the practical and theoretical material is linked. We will study notions typically thought to belong to quantum theory, such as entanglement, no-cloning, teleportation, and complementarity. It will turn out some of these notions also make perfect sense in other settings. For example, the very same categorical description of quantum teleportation also describes classical encryption with a one-time pad. We identify characteristics of classical and quantum information, aiming to equip students to choose the right tools and techniques for future problems they may encounter.

Prerequisites:

(INFR11136 or MATH08057 or INFR11188) and INFR08029

(or equivalent)

Learning outcomes:

On completion of this course, the student will be able to:

1. Identify features of current quantum programming platforms;
2. Model quantum protocols categorically and prove their correctness graphically;
3. Apply and prove basic results about monoidal categories;
4. Fluently manipulate the graphical calculus for compact categories;
5. Differentiate between categories modelling classical and quantum informatics.

Assessment:

50% written exam, testing learning objectives 3,4,5.

25% lab work, testing learning objective 1 and 2.

25% coursework, testing learning objectives 2 and 3, similar to weekly non-assessed exercise sheets.