

## Unit Title: **Fundamental Product Programming Skills**



Unit Level:	<b>Three</b>
Guided Learning Hours:	<b>60</b>
Ofqual Unit Reference Number:	<b>Y/507/3216</b>
Unit Review Date:	<b>31/07/2017</b>
Unit Sector:	<b>9.3 Media and Communication</b>

### **Unit Overview**

This unit will introduce different styles of programming or different ways of thinking about code. These different paradigms are used as design tools, the code often looks very similar at the file level but the way it is structured and the way that it interacts is vastly different. Different paradigms are used in different situations, to solve different problems and by different industries, it is therefore important for learners to have an appreciation of these different techniques for designing code.

The aim of this unit is to teach fundamental programming concepts, the language used isn't as important as the fundamental skills themselves; problem solving, code design, compilation, testing and debugging. These skills are shared across all languages and all platforms. Any C based language is viable for this unit. Learners will create their own programs to solve specific problems through an explorative approach which will help learners to retain information and reinforce it with experimentation.

---

### **Learning Outcomes**

#### ***The learner will:***

- FPPS 1** Understand how to design, develop, test and debug computer programs
- FPPS 2** Understand different programming paradigms.
- FPPS 3** Understand how to read input from the user and output a result
- FPPS 4** Be able to use control systems to design algorithms
- FPPS 5** Be able to use different types of variables to design data structures
- FPPS 6** Understand the importance of readable and maintainable code

---

### **Indicative Content**

- Designing Code: Problem solving, class diagrams, sequence diagrams, activity/flow diagrams.
- Creating Code: Editors, compilers, linkers, debuggers, interpreters, breakpoints, debug output, errors and warnings. Integrated development environments. Reserved keywords. Arithmetic. Source Control.
- Programming Techniques: procedural, object oriented, event-driven, component based and other paradigms.
- Output: Debug, command prompt, graphical, windowed, libraries.
- Keyboard: Reading key data, detecting keyboard events.
- Mouse: Polling, button data, position data, events, position.

## Unit Title: **Fundamental Product Programming Skills**



- File: Reading, writing, creation, deletion, checking existence, directories and formats. Selection statements: if, else, switch.
- Iteration statements: for, while.
- Jump statements: break, continue, goto.
- Data types: Integers, Floating point numbers, characters, booleans, long numbers and doubles.
- Data structures: Arrays, lists and classes, enumerations.
- Clear and Easy to understand code: White space, indentation, variable and function naming conventions, coding standards and commenting. Individual, paired and group work.

---

### **Assessment**

This unit is assessed using the following assessment method:

- Programming Examination

See the assessment section of the qualification specification, Assessment Pack and sample assessments for full details on the assessment.

---

### **Delivery**

This unit will introduce fundamental concepts required for programming in any industry but special emphasis should be placed on how these concepts relate back to the games development, animation and special effects industries in particular. The unit will help learners to understand key concepts that are required in order to build software programs. Throughout the unit, emphasis should be placed on experimentation and exploration of the concepts as they are introduced.

Successful teaching of this unit will be achieved with a combination of tutor-led learning sessions and individual exploration of the concepts, through playing with input values and viewing the resulting output. Sessions should be practical and the emphasis should always be on exploration and experimentation.

This unit should not be taught in the order of the Learning Outcomes. It is recommended that outputting data to the command line and selection statements are the first subjects to introduce to learners, followed by reading input, different data types and introducing iteration statements. Advanced concepts such as mouse and file input, programming paradigms and data structures should be introduced later in the course. Showing good practice for writing clear and easy to understand code should be reinforced throughout the teaching of the unit and techniques for designing, developing and debugging can be introduced as other subjects are taught. Where possible subjects which this unit should be aligned with the Fundamentals of Maths and Logic unit so that the mathematical prerequisites for a programming subject are introduced first.

This unit should be taught in a practical manner and wherever possible concepts should be introduced alongside practical uses that relate back to the games development, animation or visual effects industries. Class tutorial session should be taught with the aid of computers

## Unit Title: **Fundamental Product Programming Skills**



for each learner, so that they can follow what is being taught by creating their own programs at the same time. Tutors should then periodically ensure that all learners are following along with the material, answer any queries that they may have and encourage experimentation with the programs they have created.

Formative assessment for this unit should be practical and keeps the emphasis on experimentation, allowing learners to investigate the various subject areas which have been introduced and then present their findings utilising both written and oral communication skills.

It is recommended that the course is delivered using Python on a Windows platform using a modern integrated development environment. Working samples should be distributed to learners for more advanced topics so that programs do not need to be written from scratch each time. In some cases, broken code should be distributed to learners to help them learn to track down and remove errors in unfamiliar code.

---

### **Resources**

Learners must have access to:

- Computers with a suitable integrated development environment installed.