Cambridge Technicals Information Technology L3

Diploma - 720 GLH QN: 601/7101/7

KS5 - Preparation Evening

- Information about courses
- Preparation work for over the summer holiday
- Subject specific equipment requirements
- Exemplar work of summer task

Information about courses
Qualification overview

11 units make up the 720 GLH

1st Year:

3 Compulsory units + 3 additional units

2nd Year:

1 Compulsory units + 4 additional units

Qualification Pathway
Application Developer

There are four possible pathways to choose within the qualification:

- ❑ Data Analyst
- ❑ **Application Developer**
- ❑ Emerging Digital Technology Practitioner
- ❑ IT infrastructure Technician

Y12 Units

Fundamentals of IT	Exam (Jan)	90 GLH
Global Information	Exam (May/June)	90 GLH
Application Design	CW	60 GLH
Project Management	CW	60 GLH
Product Development	CW	60 GLH
Software Engineering for Business	CW	60 GLH

UNIVERSITY ACADEMY HOLBEACH

Y13 Units

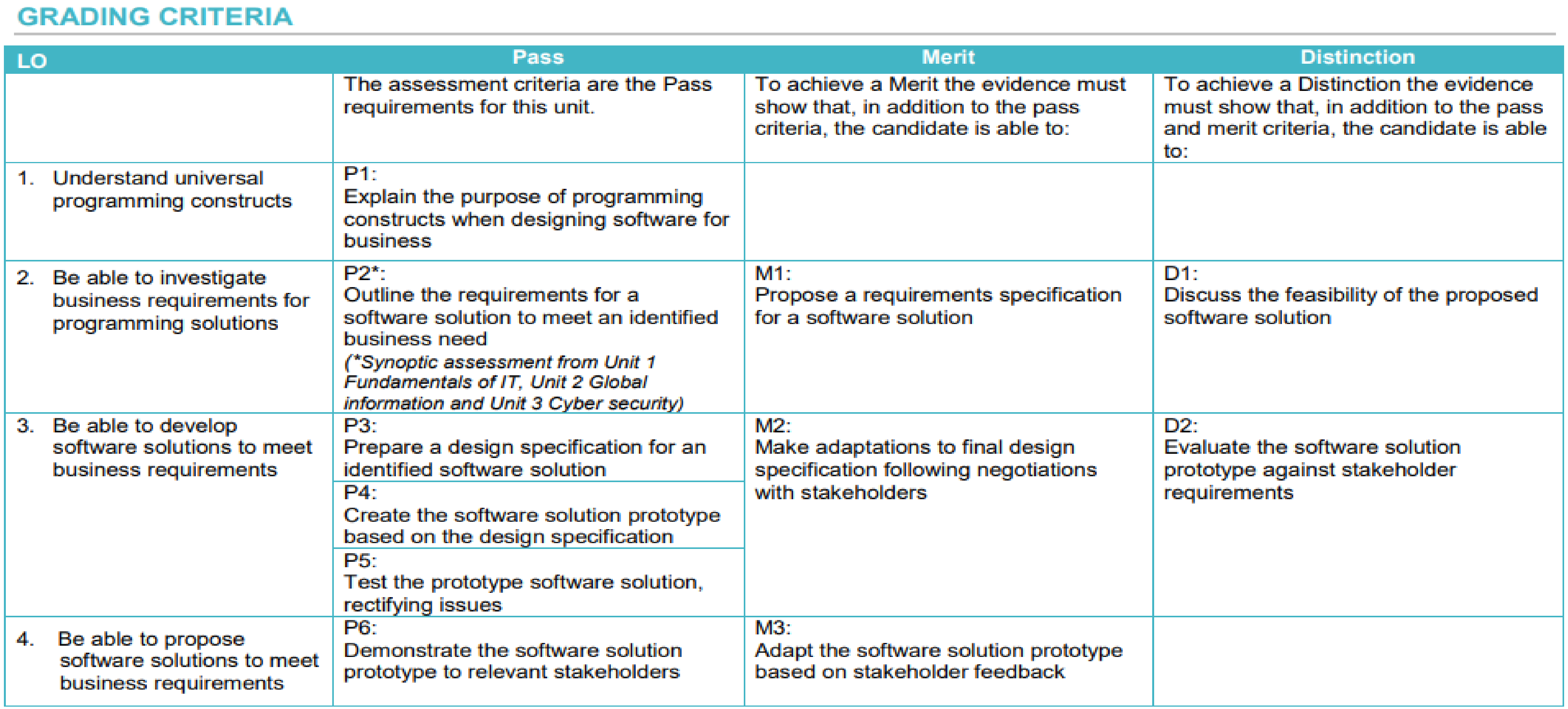
Cyber Security	Exam (Jan)	60 GLH
Mobile Technology	CW	60 GLH
Internet of everything	CW	60 GLH
Games Design and Prototyping	CW	60 GLH
Web Design and Prototyping	CW	60 GLH

Preparation work for over the summer holiday

Unit 14 – Software Engineering for Business

You are Software Engineer for Progress Computing Solutions. Part of your role is to train new Software Engineers to the company to help them understand the different programming constructs needed to create a well-designed, functioning program. This role requires you to describe a range of different programming constructs.

Unit 14 – Learning Objectives and sub topics



GRADING CRITERIA

LO	Pass	Merit	Distinction
	The assessment criteria are the Pass requirements for this unit.	To achieve a Merit the evidence must show that, in addition to the pass criteria, the candidate is able to:	To achieve a Distinction the evidence must show that, in addition to the pass and merit criteria, the candidate is able to:
1. Understand universal programming constructs	P1: Explain the purpose of programming constructs when designing software for business		
2. Be able to investigate business requirements for programming solutions	P2*: Outline the requirements for a software solution to meet an identified business need *(*Synoptic assessment from Unit 1 Fundamentals of IT, Unit 2 Global information and Unit 3 Cyber security)*	M1: Propose a requirements specification for a software solution	D1: Discuss the feasibility of the proposed software solution
3. Be able to develop software solutions to meet business requirements	P3: Prepare a design specification for an identified software solution	M2: Make adaptations to final design specification following negotiations with stakeholders	D2: Evaluate the software solution prototype against stakeholder requirements
	P4: Create the software solution prototype based on the design specification		
	P5: Test the prototype software solution, rectifying issues		
4. Be able to propose software solutions to meet business requirements	P6: Demonstrate the software solution prototype to relevant stakeholders	M3: Adapt the software solution prototype based on stakeholder feedback	

A booklet – P1

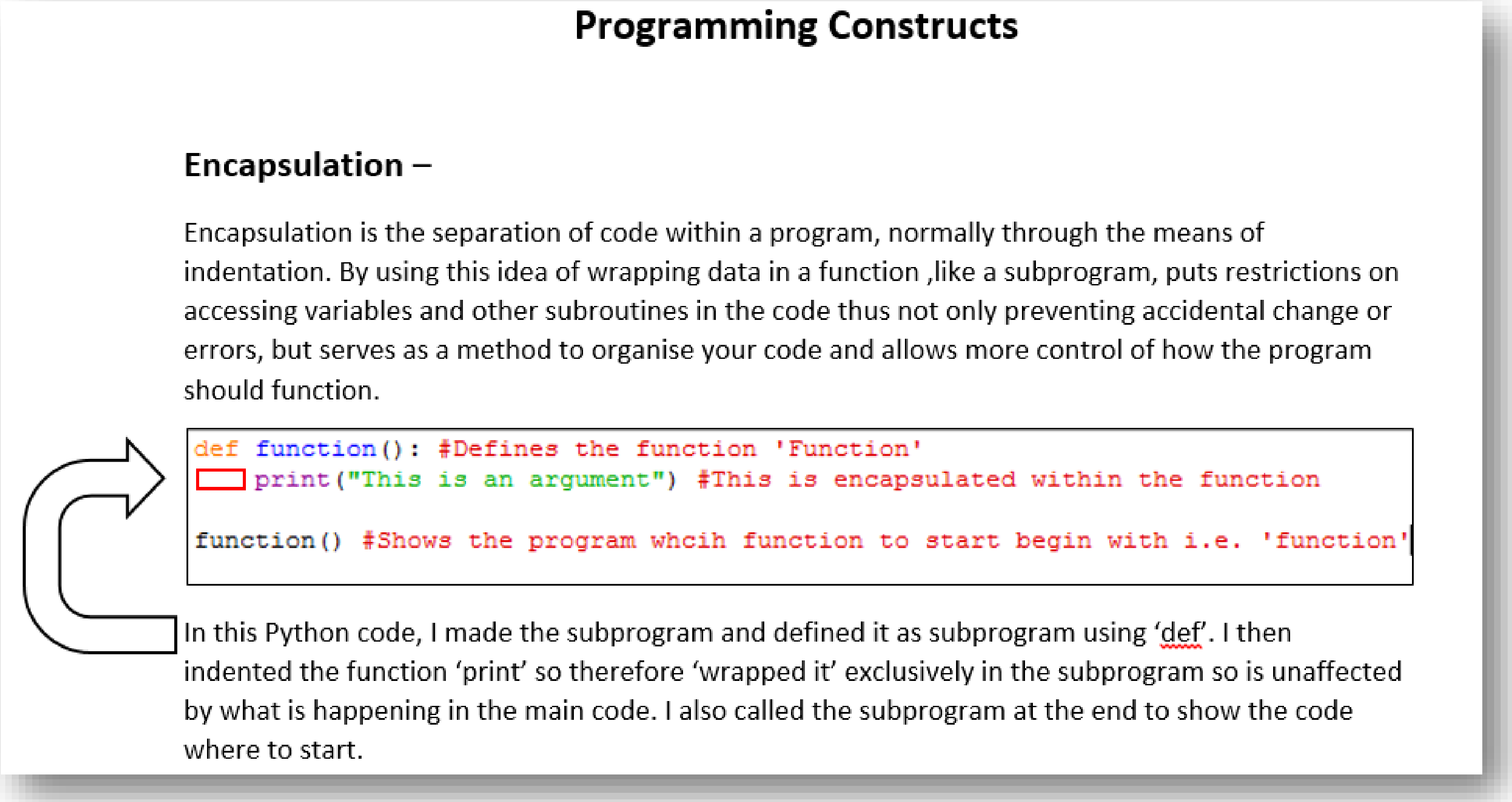
This is will be given to a trainee software developer to provide them with an explanation of the different programming constructs which are used when programming

The hand-out should cover the following Programming constructs:

- encapsulation (e.g. functions)
- predefined subroutines (e.g. string manipulation, screen output).
- the use of parameters
- declaring and using variables with fundamental data types:
    - integer
    - floating point
    - string (or equivalent e.g. array of chars)
    - Boolean
- converting between data types (casting)
- selection (if, then, elseif, else, endif)
- iteration (fixed loop, conditional loop)
- GUI objects and their different applications (e.g. text box, label, button)

The hand-out must be in your own words and provide a comprehensive summary of your findings. The hand-out must be created using appropriate titles, each construct must have its own section with a clear explanation. You should also use clear screenshots to show the construct in use.

Exemplar



Programming Constructs

Encapsulation –

Encapsulation is the separation of code within a program, normally through the means of indentation. By using this idea of wrapping data in a function ,like a subprogram, puts restrictions on accessing variables and other subroutines in the code thus not only preventing accidental change or errors, but serves as a method to organise your code and allows more control of how the program should function.

```
def function(): #Defines the function 'Function'
    print("This is an argument") #This is encapsulated within the function

function() #Shows the program whcih function to start begin with i.e. 'function'
```

In this Python code, I made the subprogram and defined it as subprogram using ‘def’. I then indented the function ‘print’ so therefore ‘wrapped it’ exclusively in the subprogram so is unaffected by what is happening in the main code. I also called the subprogram at the end to show the code where to start.