UNIVERSITY FOUNDATION PROGRAMME

COMPUTING SPECIFICATION

PREPARING STUDENTS FOR UNIVERSITY SUCCESS

FOR TEACHING FROM 2021

August 2021

CATS UFP

CATS UFP is a Level 3 course, specifically designed to help international students move successfully from secondary education to a UK University.

The CATS UFP is delivered over 420 directed hours of teaching and learning, over 3 subjects, and utilises a rigorous style of study, within a pastorally supportive and culturally stimulating environment that enables students' learning to develop and progress successfully. Students are able to access a variety of assessment methods that are common in UK Universities, such as portfolios, presentations academic posters, and examinations combined with content specifically designed to build on prior learning from courses around the world.

English for Academic purposes is an essential part of CATS UFP, and all students will take an English course that supports their learning and prepares them for university life, as well as having access to many extracurricular activities that further reinforce their use of English. Assessment design within each subject carefully focuses on subject knowledge and skills, rather than the ability to cope with English as a second language.

CATS Colleges provide a stimulating intellectual and diverse environment with small classes; thus, enabling the best learning to happen. With CATS UFP, all learning happens with teachers who have excellent subject knowledge and are expert in creating a positive learning environment for students from a wide range of backgrounds.

CATS UFP has a successful record of accomplishment and is highly respected by UK universities. With this qualification, students with 12 years of schooling from their own country can make the progression that they want, to a wide range of UK universities, including those ranked most highly for both research and teaching. CATS UFP has strong advocates in its alumni, who display what a CATS UFP qualification can give them. Graduates report that they feel very well prepared for university study; often, better prepared than students from other Level 3 programmes. Universities have confirmed this, through testimonials and through extensive consultation with university based External Examiners it has gained excellent credibility with UK universities.

WHY CHOOSE UFP COMPUTING

Dynamic and engaging content:
CATS College has offered computing courses since 1982. This experience is reflected in examination results and successful student university destinations. Our content is designed to engage students through topics and issues that are relevant today.

Real life skills:
Students will develop the knowledge and skills needed to analyse data, think critically about issues, and make informed decisions – all skills that are needed for further study and employment.

Assessment success:
UFP Computing involves a blended learning approach to assessing students that enables them to access content on our VLE and demonstrate a wide range of skills and abilities. There are 2 methods of assessment- coursework and examination papers.

- Our coursework uses a variety of assessment styles including group presentations, individual reflection, and research. Topics are contemporary and engaging to support students in developing key skills required for future University studies.

- Our examination papers use a variety of assessment styles including multiple choice and short answer so that students feel more confident and engage with the questions. Real life case studies will be used wherever possible to make it easier for students to relate to and apply their knowledge and skills developed throughout the course.

Sensitivity towards international students:
The UFP Computing course has been designed to consider the challenges that international students will face when studying a British qualification. Coursework and examination assessments are tailored made to ensure students can access, understand, progress, and achieve to the best of their abilities.

AIMS OF THE UFP COMPUTING COURSE
CATS College wants to enable students to:

- Develop an enthusiasm for studying computing
- Develop an understanding of Computing concepts and theories; and see how these relate
- to the real world of Computing.
- Be inspired, motivated, and challenged by following a broad, coherent, practical, satisfying, and worthwhile course of study.
- Provide insight into, and experience of how computers work, stimulating learners' curiosity and encouraging them to engage with computers in their everyday lives
- Allow student to make informed choices about further study or career choices.

SPECIFICATION AT A GLANCE

Unit	Content and Allocated Class Time		
Unit 1 **Introduction to Computer Science** Hardware and Software Networks and Internet	Input/output Devices		4h
	Parts of a Computer System and Storage		4h
	System & Application Software		6h
	Internet & Security		5h
	Networks		5h
	Robotics - Design and Applications		4h
	App Development, Web and Phone Coding (e.g., MIT App Inventor)		3h
	Structure of Raspberry Pi (single board computer with programming languages)		2.5h
	Summative Assessment Unit 1		1.5h
	Total		**35 hours**
Unit 2 **Data Representation; Logic and Arithmetic; Algorithms Design and Complexity; Programming Languages and Development; Relational Databases.**	Unit A	Fundamentals of programming	13h
		Structured Programming	13h
		Logic and Arithmetic Foundation of Data Representation and Programming	12h
	Exam Unit A (including tuition)		2h
	Unit B	Algorithms	15h
		Relational Databases	8h
	Exam Unit B (including tuition)		2h
	Total		**65 hours**
Unit 3 **Computing Applications and the Final Computer Project**	a. Fundamentals of Python Programming, or b. Fundamentals of SQL Programming, or c. Programmed Robots with Raspberry Pi or Arduino, or d. App development (phone or web)		30h
	Project Unit 3		10h
	Total		**40 hours**
Total	**140 hours**		

***ASSESSMENT OVERVIEW**

UFP Computing is to be delivered within 1 academic year at CATS College, full time. Students will be assessed via:

Unit 1 Summative Assessment	20%	Unit 1 will be assessed by a 1 hour 30 min exam The Summative Assessment is given 60 marks
Unit 2 Comprehensive Examinations	55%	Students take two Exams, all assessing Unit 2 (one for each Part A and B). Exam Unit A is 1 hour 30 mins in class and Exam Unit B is 2 hours in class (not including necessary tuition before the assessment). Unit 2 Part A Comprehensive exam is 75 marks. Unit 2 Part B Comprehensive exam is 90 marks.
Project	25%	Unit 3 is assessed by a computing project which is 75 marks. Preparation of the Project takes two weeks (not including the necessarily tuition before the assessment) **Unit 3 Topics are taught as preparation for the project** (a total of 30 h)

Marking

Assessment	**Marks**
Summative Assessment Unit 1	60
Comprehensive Assessment Unit 2 Part A	75
Comprehensive Assessment Unit 2 Part B	90
Computing Project	75
Total	**300**

ASSESSMENT OBJECTIVES

Assessment objectives (AOs) are designed for Level 3 Computing. CATS UFP places a strong emphasis on the use of computing in an international context when compared to other Level 3 qualifications

	Objective	Weighting
A01	Recall, select and communicate their knowledge and understanding of computer technology	
A02	Apply knowledge, understanding and skills to solve problems by using computer programs	
A03	Analyse and evaluate, make reasoned judgements, and present conclusions	

** Contains public sector information licensed under the Open Government Licence v3.0*

TIMELINE

	September start students
September	
October	Unit 1 Coursework Done Unit1 Summative Assessment
November	
December	
January	Unit 2 Part A Coursework Done First Exam Unit 2 (Part A)
February	
March	Unit 2 Part A Coursework Done Second Exam Unit 2 (Part B)
April	Unit 3 Project start
May	Unit 3 Project Done Complete the final grade
June	Results Published
July	

SUGGESTED READING MATERIALS

Revision materials

AQA Computer Science for GCSE Student Book
24 June 2016

Steve Cushing

ISBN: 978-1471866197

GCSE OCR Computer Science, Hodder Education, Paperback, 31 March 2020

S Robson, PM Heathcote

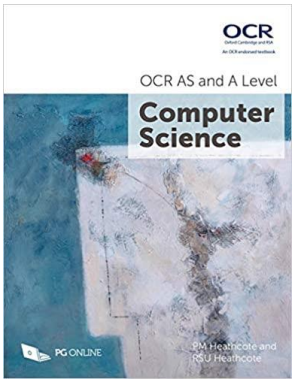
ISBN: 978-1910523216

Level 3 Texts

AS and A Level AQA Computer Science
24 April 2016

PM Heathcote & RS Heathcote

ISBN: 978-1910523070

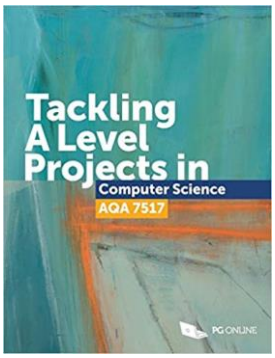


AS and A Level OCR Computer Science
12 September 2016

PM Heathcote & RS Heathcote

ISBN: 978-1910523056

Computing projects



Tackling A Level Projects in Computer Science for AQA 30 January 2020

PG Online

ISBN: 978-1910523209

Tackling A Level Projects in Computer Science for OCR 18 November 2019

Ceredig Cattanach-Chell

ISBN: 978- 1910523193

CONTACT US

UFP Chief Examiner STEM

CATS Cambridge
1 High Street
Chesterton
Cambridge CB4 1NQ

Tel: +44 1223 314431

rmathers@catscambridge.com

Director of CATS UFP

Mob: +44 7891674841

jhawkins@catglobalschools.com

SYLLABUS DETAILS

Unit 1

Input/Output Devices

- Identify input/output devices in given situation (e.g., lift or self-serve checkout).
- Identify different input/output devices and where best used – Keyboard, Mouse, MICR, OCR, OMR, Bar codes, RFID, impact and non-impact printers, monitors, e-ink, speakers.

Parts of a Computer System and Storage

- Processor, CPU: the control unit, the arithmetic logic unit, the cache, and the clock
- Buses and controllers
- Current Memory Types. RAM and ROM.
- Understand the need for, and means of, communication between computer's components.
- Factors affecting processor performance
- Primary and Secondary Storage
- Virtual storage.
- File compression.

System and Application Software

- Difference between Application Software and Systems Software.
- The need for, function and purpose of Operating Systems (OS).
- Identify the basic functions of a range of application software.
- Utility software such as disk clean-up, anti-virus/malware, disk formatting and back-up techniques
- Libraries, translators (assembler, compiler, interpreter)
- Classification of programming languages
- Open source vs closed source.

Networks

- Characteristics of networks and the importance of protocols and standards.
- Network topology
- Internet structure:
  a. The TCP/IP stack.
  b. DNS.
  c. Protocol layering.
  d. LANs and WANs.
  e. Packet and circuit switching.
- Types of networking between hosts: client-server and peer to peer.
- Wireless networking

Internet & Security

- Antivirus, Firewalls, Username & Passwords, Encryption.
- Security Policy.
- Physical/Software Security.
- Viruses, Worms, Trojans, Ransomware.
- Risks with Internet of Things (IOT).
- Difference between Crime and Malpractice. Internet Ethics.
- Types of Backups.
- Creating a backup policy.

Control Systems and Robotics

- Understand key concepts relating to robots and robotics systems
- Identify examples of robots and the main parts of a robot and their function
- Understand the elements of a simple control system and how to test it
- Understand basic programming concepts and visual programming language
- Set up a robot, implement robotic motion and control a robot in an environment

App Development

- What is an App? What is App Development?
- Visual programming environments
- App Prototypes: why make an App Prototype?
- How can I commercialise or publish a Mobile Application?
- MIT App Inventor (https://appinventor.mit.edu/)

- Blocks-based coding programs

Comparing Raspberry Pi and Arduino

- Structure of Raspberry Pi
- How to connect Raspberry Pi to external devices such as monitors, keyboard, sensors, etc.
- Structure of Arduino
- Arduino products, including Boards, Modules, Shields, and Kits.

Summative Examination (1h 30 min and 60 marks)

UNIT 2

Part A

Fundamentals of Programming

- Data types: integer, real/float, Boolean, character, string, date/time, records, arrays, lists, sets
- Programming Concepts: variable and constant declaration, assignment, iteration, selection, subroutine (procedure/function)
- Arithmetic operations in a programming language: addition, subtraction, multiplication, real/float division, integer division, including remainders, exponentiation, rounding truncation
- Relational Operations in a programming language: equal to, not equal to, less than, greater than, less than or equal to, great than or equal to
- Boolean operations: NOT, AND, OR, XOR and XNOR and truth tables
- String-handling operations in a programming language: length, position, substring, concatenation, character -> character code, character code -> character, string conversion operations
- Files and Exception handling
- Local/Global Variables in Subroutines/Programs

Structured Programming

- Hierarchy or Structure Charts
- Flowcharts (sequence, assignment, selection, iteration)
- Pseudo-code (sequence, assignment, selection, iteration)
- Naming conventions, code layout and comments
- Dry Runs and trace tables
- Use of an IDE to develop/debug a program

Logic and Arithmetic Foundation of Data Representation and Programming

- Units of Information: Bits and Bytes. Multiples of Bytes and International System
- Binary Numbers, Hexadecimals Numbers, and other number systems. Representation of negative integers or decimals (floats)
- Logical gates: circuits and logical tables. Universal Logical Gates.
- Decision trees: concept and usage
- Encryption: modular arithmetic, shifting the alphabet and standard ciphers

Comprehensive Exam (2 h and 75 marks)

PART B

Algorithms

- Definition and properties (Input, Output, Definiteness, Finiteness, Effectiveness)
- Implementation of algorithms
    a. Data types and structures in algorithms (Arrays, Sets, Dictionaries, Records, Files)
    b. Algorithm specification (Standard Algorithms, Input Validation)
    c. Analysis and Design (Input, Output, Process, use of Pseudocode/Flowcharts)
    d. Testing and Evaluation
- Iterative and Recursive algorithms
- Standard algorithms (bubble sort, insertion sort, binary search, and linear search)
    a. Introduction: definition and concepts
    b. Implementation of sorting and searching algorithms
    c. Comparisons (complexity, sensitive cases)
- Big-O notation, functions, time complexity

Relational Databases

- Relational Databases
- Flat file, primary key, secondary key, foreign key, entity relationship modelling, normalisation, and indexing

- Methods of capturing, selecting, managing, and exchanging data
- Normalisation to 3NF
- Introduction to Structured Query Language (SQL)
    a. Relational Database Management System (RDBMS)
    b. SQL Server/Client for beginners
    c. Queries, Tables
    d. Create, Read, Update, and Delete (CRUD)

Comprehensive Exam (2h and 90 marks)

UNIT 3

Note: The following topics are for student tuition depending on their choice of the project. These topics are assessed with the project.

Fundamentals of Python Programming

- Data types, operators, and I-O
- String and Numbers
- Selection
- Iteration
- Lists, Tuples, Dictionaries, Arrays and Sets
- Validating user input
- Searching and sorting
- Functions
- Reading and writing files
- Useful Python's packages: Numpy, Matplotlib, Itertools, Pandas
- Program design
- Testing and Debugging

Relational Databases. Structured Query Language (SQL)

- Introduction to SQL
- Defining and Updating Tables with SQL
- Python packages (e.g., SQLite) for relational databases (e.g., SQL, MySQL)

Robotics with Raspberry Pi or Arduino

- Arduino vs. Raspberry Pi for Robotics
- Arduino Tutorials and Projects
- Raspberry Pi Tutorials and Projects
- Assembling a robotic vehicle and learning how to drive each component
- Practical and theoretical elements of robotics
- Learning to manage complex mazes

App Development

- App Development with MIT App Inventor

COMPUTING PRACTICAL PROJECT (15H AND 75 MARKS)

- First week: students choose the problem and prepare a draft with their first results on researching the problems
- Second week: students choose the appropriate design and starts their technical solution
- Third week: testing and evaluation, checking last points and submitting the project
- Unit 3 continue with topics tailored for student projects and include the tuition

COURSEWORK AND ASSESSMENT DETAILS

Problem solving, practical and research skills are assessed using the Summative Assessment, two Comprehensive Exams (Skill and Knowledge Assessing) and one Computing Project at the end of the year.

Students will sit all, and all will be contributing to the final grade.

Unit 1 is assessed through **Summative Assessment** (60 marks suggested, or up to **20 %** of final grade)

Unit 2 is assessed by two Skill and Knowledge Assessments (up to 55 % of final grade):

1. **Comprehensive Exam 1** at the end of Part A (75 marks suggested, or up to **25 %** of final grade)
2. **Comprehensive Exam 2** at the end of Part B (90 marks suggested, or up to **30 %** of final grade)

Unit 3 is assessed via a **three-week Project**. (75 marks suggested, or up to **25 %** of final grade)