

Computer science

Contents

Course information

Expectations	3
Overview of Year 12 & Year 13	4
Unit 1: Computer Systems	6
Unit 2: Algorithms and Programming	8
The Project	10

Study skills and revision tips

Study Skills: Lessons	12
Study Skills: Active Revision	14
Exam Preparation	16
Useful Resources	18
Dealing with distractions	19
Beyond the Specification	20

CONTACT DETAILS

- 00
- pbr@lpgs.bromley.sch.uk
 https://lpgs.fireflycloud.net (course resources)

Expec tations

This list may seem basic and full of obvious points, however in year 12 students are given a little more freedom than they are used to. This can mean some students forget the fundamentals of being a successful student.

As a computing student you are expected to:

- Attend all lessons
- Be punctual to lessons
- Bring your Computing folder to all lessons
- Maintain an organised set of notes
- Revise regularly
- Be aware of your strengths and weaknesses
- Work on overcoming your weak areas
- Complete all homework to the best of your ability
- Complete at least one practical exercise a week
- Ask if you are unsure of anything



CHALLENGES you will face

The volume of new terminology you need to understand

This will be the main challenge you will face. You can help yourself by reading appropriate chapters/powerpoints **before** the lesson so you are familiar with the words that will come up in the lesson. **Regular revision will also be key.**

Juggling multiple subjects

Although you are studying fewer subjects the demands on each will be high. Ensure you are **organised and disciplined** enough to keep up with the workload for all subjects.

Overview of year 12 & Y13

From the start of Y12 you will be preparing for your final exams that will be taken at the end of Y13

The course is broken down as follows:

Assessment: 2 exams 2.5 hours each and 1 piece of coursework

01: Computer Systems	40%
02: Algorithms and Programming	40%
03: Programming Project	20%

Practical skills

During your first year you will learn to program. This will be a very practical process where you will be using the computers to write your programs and run them. It will be tough going at first and you will have to hone your problem solving skills, but it will be very rewarding when you get your programs running!

You will complete some programming tasks in pairs as well as independently.

You will also watch live coding demos and be expected to ask questions and complete partially completed code.

Programming and planning algorithms is a skill required for both unit 1 and unit 2 as well as the project.

Theory

As well as programming you will also be required to learn a lot of theory in the form of definitions, descriptions and explanations.

Our aim will be to complete the majority of unit 1 by the end of YEAR 12, some of this will overlap with content needed for unit 2.

In year 13 we will need to complete any remaining theory from unit 1, complete unit 2 whilst also completing coursework. Unit 2 actually has less content than unit 1 and we will have covered some areas via the programming tasks in year 12 as well as some topics that feature in both papers.

The coursework will require a significant amount of work being completed outside of lessons. Your challenges will include juggling the coursework and preparing yourself fully for the 2 comprehensive exams.

ASSESSMENTS across y12 & y13

Throughout the 2 year you will have multiple assessments including:

- September: Baseline Test
- Half termly exam style assessments
- Online review quizzes
- Programming exercises
- Weekly revision quizzes in Y13

COMPUTER SYSTEMS

1.1 Components of a computer and their uses

You will learn about the main components found within a processor and understand what happens when an instruction is sent to the computer. You will compare RISC and CISC processors as well as multi core and parallel systems.

You will look at a range of input, storage and output devices and be able justify their choice in a range of scenarios.

1.2 Software and software development

You will understand the role of an operating system in the running of a system including scheduling, memory management, interrupts and translators.

You will also cover the core concepts involved in procedural programming including variables and constants, procedures and functions and a range of string handling and file handling processes. You will also learn a second programming paradigm called assembly language.

1.3 How data is exchanged between different systems

This sections covers databases, networks and web technologies. You will learn about flat files databases and relational databases including a range of terminology such as primary and foreign keys.

You will look at the characteristics of networks including the internet. You will cover range of network specific terminology including TCIP/IP stack, DNS, protocols, packet and circuit switching.

You will also learn some basic HTML, CSS and JavaScript

1.4 Data types, data structures and algorithms

This section covers a range of programming concepts such as data types, binary arithmetic, arrays, stacks and queues as well as boolean logic.

It will require you to apply your understanding to specific problems rather than just explain how something works.

1.5 Legal, moral, ethical and cultural issues

Technology has brought about all sorts of new issues that didn't exist before. This section will give you an awareness of 4 key laws that relate to technology

- The Data Protection Act 1998
- The Computer Misuse Act 1990
- The Copyright Design and Patents Act 1988
- The Regulation of Investigatory Powers Act 2000

It will also make you think about the moral, social and ethical opportunities and risks associated with technology. This will cover issues like:

- Computers in the workforce
- Automated decision making
- Artificial Intelligence
- Environmental effects
- Monitoring Behaviour
- Piracyand offensive communications

SAMPLE QUESTIONS

from past papers



Algorithms and PROGRAMMING

2.1 Elements of Computational Thinking

You section looks at 4 strands of computational thinking:

Thinking abstractly—removing the parts of the problem that are irrelevant leaving you with the core details that will affect your solution

Thinking ahead: We look at the sort of planning involved in creating a system including consideration of the inputs and outputs.

Thinking procedurally: Determining the steps needed to solve a problem and breaking these sown into sub procedures.

Thinking Logically: Determining when decisions need to made and the logical conditions that affect the outcome of a decision

2.2 Problem solving and programming

We will examine how computers can be used to solve problems and programs can be written to solve them.

Think includes key programming terminology and concepts including programming constructs, variables and modular design using functions and procedures.

You will also look at the process involved in developing software from understanding what the problem is through to designing, building and testing.

2.3 Algorithms

An algorithm is a set of instructions that perform a specific task. You will practice writing algorithms for a given problem and look at some commonly used algorithms including:

- Bubble sort, insertion sort,
- Binary search and linear search
- Adding and removing data from queues and stacks

SAMPLE QUESTIONS from past papers

Describe what is meant by the term (Integrated Development Environment Identify and describe three fe commonly found in an IDE that will programmers to find any bugs in code. [6]		n IDE nt [2] atures II help their	Compare a program procedure with a function [2]
	FUNCTION Mys IF numb RE ELSE END IF END FUNCTION	stery (number wer < 10 THEN TURN number TURN Mystery () (numbe r -9)
	Identify the para Describe the log	ameter that is passe gical decision that i	ed to this procedure [1] s made [2]
	The programme the output that	ertests this code wit would be obtained	th the value 16. Show . [5]
Explain why it is necessary for the developers to use abstraction [2]			
Explain how a map is an example of an abstraction [2]			
Explain how levels of abstracti on assist the map user [3] As the size of an array increases the average			creases the average
	numbe grows logaritl	r of checks require logarithmically. St hmic growth.[1]	d in a search algorithm ate what is meant by
	Assum a linea binary	ing an array is sorte r search would pei search. [1]	ed give a situation when form better than a

The Project

You will start the project in June (after you have finished your year 12 exams) and continue working on it through most of year 13. (The deadline for the coursework is normally Mid May).

The project will involve you working through the systems life cycle from defining what you intend to create, to analysing, designing, building, testing and evaluating your final system.

CHOOSING A SUITABLE PROJECT

You can make any type of project that interests you but you need to have some sort of 'end user' that will help you come up with ideas. This could be someone that runs a business, a teacher or a group of students.

Your main limitations when choosing your project are:

TIME

You will have limited time to complete your project so you must choose a project of a suitable size. If your program is unfinished by the deadline you will lose significant marks.

PROGRAMMING CONFIDENCE/ABILITY

You will all end year 12 with varying degrees of confidence in your programming ability, the project is a great way to improve your skills and learn new techniques but you should be realistic about how complex your program can be. Again if you set out to complete something that is complex but only manage to make parts of it you will lose significant marks.

Types of programs that can work well include:

- Math based quizzes
- Revision tools / quizzes for other subjects
- Stock / Order / Invoice management system
- Progress tracker for students
- Simple games



Science quiz where students had to recall information about different elements.



Maths program f or primary students learning their multiplication tables.



Stock management system for a bakery allowing them to analyse current and past sales.



Touch typing program for year 7 students with a ninja When theme. students hit different targets they are rewarded with ninja characters.



Study Skills

LESSONS

Attending lessons will be key. However you will need to do more than just show up.

BEFORE LESSONS



Read over what we will be covering in the lesson.

For many lessons I will ask you to watch a video or read through some slides that will link to the up coming lesson. This will mean you can immediately play an active part in the lesson and will already have a few questions on hand that will further your understanding. This will be essential in allowing us to complete the course in the permitted time

DURING LESSONS

The challenge is to remain focussed so you are registering what is being said. To avoid tuning out try to:

Listen carefully

Don't just tune in and out, it won't be long before you are lost. Don't sit next to someone that might distract you.



Take Notes

This is a more active activity and should help maintain your focus.

Ask questions when you don't understand.

It is impossible to stay focussed if you do not understand what is being said.

AFTER LESSONS

You need to work on retaining what is covered in lessons. How much time you spend outside of lessons has a major impact on results.

Read over notes

This will help you remember the finer details.





Write out comprehensive notes

Don't start summarising things too early or you may overlook bits.

Complete homework This will helpcheck your

understanding.





Do questions on topics that were covered earlier in the year

Don't wait until just before the exams to review things covered monthsago.

Email your teacher

If there are any parts that you do not understand do not overlook them— get help.

Study Skills **ACTIVE REVISION**

Revision is far more than sitting and reading through notes. This is a very passive approach that can be very ineffective. More active and effective suggestions can be seen below.

LITTLE and OFTEN

OTHER IDEAS

Get family and friends

to testyou

Create revision posters

and stick them on your

Make electronic flash

cards using free apps like cram or flashcards+

wall. Read a couple each time you walk into your room

Record

back

yourself and listen

1!!@#*

Synthesis

class notes on paper. Use colour and diagrams.

0



2. Rewrite these notes to be more

3. Rewrite the notes onto revision cards or posters to put on your wall.



4. Test your knowledge using your revision cards or posters.

PRACTICE

1. Attempt every past paper question on your focus topic





2. Check your answers against the mark schemes. Be harsh!

3. Identify the areas you are weak on and write out the correct answers. Spend time learning these areas by reading and writing fresh notes.





4. In a months time come back to the topic and repeat all the steps.

TAKE RESPONSIBILITY

Find the time It is up to you to find the time to ensure you are able to complete homework, synthesise & practice to the best of your ability.

Identify your weaknesses and work on overcoming them Don't ignore the areas you find hard or don't like doing. These are the areas you should be spending the most time on until you feel comfortable with them.

Be honest with yourself Are you working as hard as you can? Are you marking work accurately? Don't tell yourself everything is fine if it is not. Seek help early on if you are struggling.

Exam PREPARATION

You will ultimately be assessed on how well you perform in your endof year exams. Do not leave your preparation to last minute, get organised early so there are no last minute panics. You want to spend the weeks leading up the exams revising not getting organised and finding things.



Final tips

WHERE you study is just as important as how you study

You will need to find places **IN and OUT of school** that you can study. It is recommended that this area is a study area only and not used by you for other purposes.

Nobody is a revision robot

Take short breaks every hour or so to give yourself a rest. Drink water and eat healthy snacks to keep your brain ticking over.

Focus on understanding

There will be some definitions that you will have to learn accurately but really the most effective way to revise is to concentrate on understanding rather than memorizing. Memorising requires huge efforts and can limit the sort of questions you can answer whereas if you have a good understanding of a topic you should be able to answer a broad number of questions on a topic.

Make your notes easy to follow

Make sure you can link your notes back to the specification with simple labelling. This way you know which bits you have covered and what you have to focus on next.

Make your notes visually appealing

Take time with your notes so you can read them back easily. Use a bit of colour! colourful notes are easier to memorise than plain black and white ones. Use diagrams where possible to break up the text and don't overload each page, perhaps use revision cards to break up topics.

Do plenty of past papers / questions and mark them

This will get you familiar with the layout and structure of the questions. Understanding how to answer a question fully is just as important as understanding the theory. Using the mark schemes to mark your own work will improve your technique.

Reward yourself

Give yourself something to look forward to after revision each day. People who manage to find the right balance between study and leisure are the ones who get the topmarks.

Think positively

If you have done the work you will have nothing to worry about when the exam comes around.

USEFUL RESOURCES

Websites

https://lpgs_fireflycloud_net

This site contains all the class notes, practice questions, past papers and marks schemes. Familiarise yourself with it early—favourite the link.

Youtube_com-search for 'OCR Specification Order'

A large number of short videos covering the key concepts for Y12. A great way to refresh your memory at the end of the day.

https://isaaccomputerscience_org/

A Level Computer Science content and quizzes. Look out for other opportunities including conferences to attend.

Flashcard Apps

Flashcards+

Simple but effective flashcard app that lets you organise your cards into those you are confident with and those you need to revise.

CRAM_com

Make flashcards quickly and view cards made by others. Includes a range of different ways to use the cards including quizzes and games.

Quizlet_com

Similar to Cram in that you can view cards made by others and use the cards to complete quizzes.

Other apps

ExamTime_com

Access to a range of study tools including flashcards, quizzes, Mindmaps, notes and a study planner

SimpleMind+

Great mind mapping software to organise concepts in a different way.

Dealing with distractions

Can't keep away from



Try the following:

Go to the library to do your work and do not take your phone, laptop, tablet with you. Get someone you trust e.g a parent to change your password to instagram etc. They can give you access back at the weekends. This is great if you struggle to be self disciplined.

Keep your phone in a different room when you are doing work. **Uninstall apps that distract you** and reinstall when the exams are over.

Get distracted on your computer?

Trying to do coursework online but keep getting sidetracked by cat videos and news sites? The resources below work by restricting the amount of time you can spend on time-wasting websites. Once your allotted time has been used up, the sites you have blocked will be inaccessible for the rest of the day

Minutesplease_com

Block a site within seconds

StayFocusd extension (Chrome web browser only) Download for free and set sites you want to block or allow.

http://getcoldturkey.com/

Cold turkey lets you block sites and programs on your computer from half an hour up to a full week

SelfControl

For apple devices—block sites for a set amount of time. Even if you delete the app the blocks will remain in place!

Memory fade

Before you have even left the classroom you will have forgotten a significant proportion of what you have seen and heard. Without revision within 24 hours you will have forgotten most of it. The steepness of the curve varies between individuals but the trend is the same. This is why regular revision is so vital.



Fighting memory fade

Do not fall into the trap of revising something once and thinking you are done with that topic. Continual revision is key. The following is recommended to truly embed something into your long term memory.

Revise new content after:

- 30 minutes (end of day)
- 24 hours
- 1 week
- 1 month

This may be hard to stick to but if you can do any sort of staggered revision like this it should help.

BEYOND THE SPECIFICATION

For some students the only time they think about computing is during a lesson and maybehomework.

If you are serious about the subject you should go beyond the concepts covered in the specification. This might spark an interest in a particular area of computing and will make you more knowledgeable when applying for universities.

building logic & problem solving skills

http://projecteuler.net./

Project Euler is a series of challenging mathematical/computer programming problems that will require more than just mathematical insights to solve. Although mathematics will help you arrive at elegant and efficient methods, the use of a computer and programming skills will be required to solve most problems.

https://www.codecademy.com/

This is a great website with loads of short and longer projects to help you develop your programming skills. Signing up is free and you can pick and choose what exercises to try.

http://www.pythonchallenge.com/

Features riddles that ask you to write small Python programs to solve. The difficulty level gets progressively harder and more cryptic. This is an excellent site for programmers/developers that like problem solving!

http://www.mindcipher.com/

this site doesn't require you to do any programming. It does, however, include the "*world's greatest brain teasers, logical puzzles and mental challenges*", so if you're just out for a quick mental workout this is the place to go!

Books of interest

When you write your personal statement it is good to mention books you have read to further your knowledge of the subject. These books are recommended by the top universities. Find more on Firefly



The Code Book by Simon Singh. ISBN: 978-1857028898

Not strictly about CS but an interesting introduction to code-breaking and cryptography, fields that have a strong connection to CS ..

The New Turing Omnibus by A Kee Dewdney. ISBN: 978-0805071665 Mini articles on Computer Science topics.

Algorithmic Puzzles by Anany Levitin and Maria Levitin. ISBN: 0199740445

Computer Science: An Overview by J. Glenn Brookshear. ISBN: 978-0321544285

Overview of what CS is all about: each topic includes historical perspective, current state, and future potential, as well as ethical issues.

How to think like a mathematician, Kevin Houston, Cambridge University Press, 2009, ISBN 978-0-521-71978-0.

Don't be misled by the title; this book is absolutely relevant for us

The Pleasures of Counting by Tom Kôrner. ISBN: 978-0521568234 Puts Maths into the context of how it is used to solve real-world problems.

Other interesting reads

- **Ghost in the Wire:** Story of prolific hacker, Kevin Mitnick
- Hackers: Heroes of the Computer Revolution—Steven Levy
- **The Cuckoo's Egg**: Tracking some of the first hackers—Cliff Stoll
- **Kingpin**: Gripping story based on a hackers exploits -Kevin Paulson

Films

Below are a selection of movies that cover a range of topics from the history of the computer and key players as well as ethical and moral issues brought about by technology. (Beware, not all are 100% accurate!)



Title	Themes
HACKERS	Hacking, computer viruses, moral issues presented to programmers
ANTI TRUST	Open source VS corporation owned technology
THE NET	Identify theft
IROBOT	Artificial Intelligence, should we develop AI basd robots?
WARGAMES	A teenage hacker almost starts WWIII. Dated but a classic
FREEDOM DOWNTIME	Based on legendary hacker Kevin Mitnick
THE STARTUP KIDS	Find out how the founders of vimeo, soundcloud, dropbox etc got started
TRANSCEDENCE	Artificial Intelligence, should we upload our consciousness into computers?
THE IMITATION GAME	Alan Turing and co cracking the enigma code
SNEAKERS	Cryptography, government and espionage
PIRATES OF SILICON VALLEY	History of Microsoft and Apple
UNTRACEABLE	Explores the macabre side of the internet
TRIUMPH OF THE NERDS	An interesting insight into the history of computers

CONTACT DETAILS

pbr@lpgs.bromley.sch.uk https://lpgs.fireflycloud.net (course resources)