



Course Outline and Essential Information

Programming Fundamentals 2 Semester 2 - 2025 - 2026

BSc(Hons) in Computer Science,
BSc(Hons) in Computer Forensics and Security,
BSc(Hons) in Physics with Modern Technology.

Module Lecturers	
	
Dr Siobhán Roche, siobhan.roche@setu.ie	Mairéad Meagher mairiad.meagher@setu.ie
Department of Computing and Mathematics	

Contents

1	Module Name	3
2	Lecturers	3
3	How to reach us	3
4	Learning Technologies	3
5	Module objectives / Learning outcomes	4
6	Assessment Breakdown	5
7	Submission and Marking	6
8	Academic Integrity	7
9	Important note about engagement in the module and time management	8

1 Module Name

Programming Fundamentals 2

2 Lecturers

For semester 2, 2025 - 2026, we switch things around a little bit:

Siobhán Roche, siobhan.roche@setu.ie, Lecturer in Department of Computing and Mathematics.

Siobhán will now take lectures and labs for groups W1 and W2, and Physics for this module.

Mairéad Meagher, mairead.meagher@setu.ie, Lecturer in Department of Computing and Mathematics.

Mairéad will take lectures and labs for groups W3 and W4 in this module.

3 How to reach us

- The quickest way to reach us is via Slack here.. We will be using this as the main form of 'outside class' communication for this module. (This is the same channel that we used last semester)
- You can also reach us via email: (see above).
- We are available during work hours from Monday to Friday, 9am to 5pm. You can email/Slack us outside of these hours and we will reply as soon as we can, but always within three days. (If this does not happen, assume your contact has gone into spam etc. and please re-contact us.) When emailing us, please indicate what module you are taking as well as the nature of your query in the subject line, and do not forget to use an appropriate greeting and sign off. It's important to be polite and to treat one another with respect; let's start as we mean to go on.

4 Learning Technologies

In order to maximise our ability to connect with one another and to make our classes as accessible as possible for everyone, we will use both synchronous and asynchronous learning technologies during this module. They will include:

- (2)lectures , (1)tutorial and (2)lab sessions per week.
- **Moodle** - our learning management system, where you can find links to the notes and a todo list each week. The link to this course is [here](#). Each

week you will be given a list of tasks that you should have completed each week.

- **tutors** - We continue to use tutors for this module. This static website will hold all the notes, labs and links to videos (link is here). This site will be updated each week to include the following week's material.
- **GitHub Classroom** - We will use GitHub Classroom for all assignments.
- **Slack** - We will use Slack for all communication within the module. (see 3 for details).

5 Module objectives / Learning outcomes

On completion of this module students should be able to:

1. Apply core problem solving approaches suitable to the programming discipline to build algorithms.
2. Write small applications using standard sequence, conditional and iterative control structures.
3. Modify and expand small applications.
4. Write small applications that use simple UI, computation and data structures.
5. Develop techniques to effectively test, debug and document small applications.
6. Analyse and explain how the above applications work.

The full module descriptor is available on the Introduction page on Moodle. Each week, we will have 2 lectures (one hour each), one tutorial hour and one 2 hour lab where we will practice the topics introduced in lectures. Usually, we will have the lab material covered before the lab times.

6 Assessment Breakdown

The assessment in this module is made up of one *formative* and two *summative* assessments.

WEEK		Class	Assessment	Tutorial attendance 10% overall
1				
2				
3				
4				
READING				
5				
6			in class written test 40%	
7				
8				
9				
EASTER1				
EASTER2				
10			Programming Assignment 50%	
11				
12				

Figure 1: Schedule of assessment

Your module assessment is made up of the following:

- **Attendance at tutorials (10) - 10%.** You are asked to attend your tutorials where written problem sheets will be given and so that you can practice any new concepts. The tutorial will usually take place in the first non-lecture hour of the week (i.e. before you start your labs)
- **Written in class test - 40%.** This will be a test based on the material covered in the first half of the module and largely based on tutorial sheets and labs.
- **Programming Assignment - 50%.**
- You will be asked to solve a problem using inheritance, etc. The problem will be fully specified and you will be asked to implement it in **Java** using the **IntelliJ** environment. You will also be asked to use GitHub¹ Classroom to (regularly) submit your code.

¹<https://classroom.github.com/>

Assignment	Formative Assign	Inclass written test 1	Prog assign 2	Tutorial attendance 3
Handout	05/01(Week 1)	02/03 (Week 6)	13/04 (Week 10)	Weeks 1 - 12
Deadline	18:00, Friday 30/01 (Week 2)	02/03 (Week 6)	18:00, Sunday 01/05(Week 12)	Attendance taken in class
In class test	none	none	Week 13	
Interview	none	Week 10	Week 13/14	
Percentage	0%	40%	50%	10%

Table 1: Full assessment schedule

7 Submission and Marking

- The **formative assignment** is to be submitted via Moodle 18:00 on Friday, 30th January, 2026 (Week 2). For this you are asked to port your Semester 1 programming assignment from BlueJ to IntelliJ. Feedback will be given to you in lab. You can use this as an opportunity to clarify any issues you may have had with the assignment (Semester 1 programming assignment) or any other issues. There are no marks for this assignment.
- The **first summative assignment** (in-class written test **worth 40%**) will take place during class time in Week 6 (date to be confirmed when the timetable is settled). Your results will be made available to you on Moodle as soon as possible after the test, usually within a week. This in-class test will be based on the material covered in the first half of the module and largely based on tutorial sheets, quizzes and labs.
- The **second summative assignment** (programming assignment **worth 50%**) will be due at the end of Week 12. As per Semester 1, you will be asked to attend a written test on the assignment in early Week 13, (time and date to be confirmed nearer the time). You are asked to make yourself available to be interviewed on your submission after the test. If we ask you to attend an interview, you will be given a time slot in Week 13 and it is your responsibility to attend at that time in person.

Failure to attend the interview (if we ask you to do so) will result in a zero multiplier being used, resulting in an overall assignment mark of 0.

In the case of the this (second) assignment, the marks are not published as the final mark is overseen by the external examiner and the finalised mark is released only after the examinations are fully processed.

- **Tutorial attendance (10%)** will be recorded each week in your tutorial class. The best 10 out of 12 weeks will be counted.

For the programming assignments, a marking scheme will be published with the specification of the assignment. As always, **be sure that you are aware of the marking scheme**. If there are marks going for a particular part, and you haven't attempted that part, there is nothing We can do. Always make it easy for the examiner to give you marks!

If you wish to seek an extension for an assignment, you must do so in sufficient time (i.e. not on the day of submission, and not when the submission date has passed) and must provide a valid reason for seeking the extension.

8 Academic Integrity

The School of Science and Computing at South East Technological University are committed to maintaining the highest standards of academic integrity. Academic misconduct, including, but not limited to, cheating may result in a mark of zero for the assignment as well as disciplinary action. Additional sanctions may be imposed depending on the case. You are responsible for ensuring that you do not get involved in cheating of any kind.

With regard to the (summative) programming submissions, we will ask you to either:

- sit a written (closed-book²) test on the submission. Depending on the results of the test, we may ask you to be interviewed on the submission. This **will** happen for the second (final) assignment. If you are not asked to be interviewed, you will be given full marks for your interview mark.
- usually we can facilitate early interviews for early submissions. If you submit early, you may choose to be interviewed early.

Your attendance for either option is mandatory and failure to attend will result in a zero multiplier being used, resulting in an overall assignment mark of 0.

The interview is to ascertain that the work is your own and that you fully understand how it works, in its elemental parts and how it works together.

We will always encourage you to work in collaboration with your fellow classmates. But please be careful not to cross the line between collaboration and using someone else's work. Please do not be tempted to use this route. It is too risky and the penalty can affect your academic future.

Due to prevalence of AI Bots in the general ecosphere, we now need to reiterate that use of AI Bots must be used with care. It is not allowed, as usual in these cases **without** referencing. Use of AI Bots (where referenced), even with good understanding will result in significantly lower marks.

Use of helpers like CoPilot are easily available but it is imperative that you do not simply accept suggestions without fully understanding them. If you do, this should be clearly referenced. You should also be able to show how you would apply such a suggestion in a different context.

²a closed book test is one where you do not have access to any online or offline resources

9 Important note about engagement in the module and time management

Part of active engagement for any module involves a degree of time management. As part of this module, we will be asking you to complete tutorial sheets, labs and quizzes between class times. These will not be graded but, by engaging in these tasks at the time, you will be in a better position to understand the next part of the module. We will approach the module in a step-by-step manner, so opting out at any part will make it more difficult for you to keep up. This is where time management will come in - you need to be careful to ensure that you keep a balance between modules.

Always ask questions, either in class or during labs. One way to help to stay engaged is to ask questions if you don't understand what is going on. Remember, when you are asking questions:

1. Just the process of asking a question means that you have learned something.
2. If you cannot understand, in most cases, you are not the only one.
3. Asking questions means that the pace of the lecture/ labs will suit you better - We will always keep going if there are no questions!