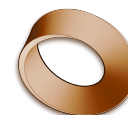


Y8A Design — MYP

Course Expectations • 2026–2027



Teacher	Mr. Alex Rawson • a.rawson@stleonards-fife.org
Room	PH1
We meet	Week A • Friday • Period 5 (14:05–15:00) Week B • Monday • Period 3 (11:15–12:10)
Course page	rawsonvault.com/myp/design

1 What this course is

MYP Design is the computing course — you will hear me call it computer science, because that is largely what it is. Most of the year goes on **robotics**: you program a TPBot, a small robot car driven by a BBC micro:bit, to solve control problems that get steadily less forgiving.

The point is not the robot. The point is the design cycle: work out what the problem actually is, design something, build it, then test it honestly and say what you would change. That loop is the whole subject, and it is why a robot that fails in an interesting way can score better than one that works by accident.

We meet once a week, so we don't have much time. Please be on time and let's make the most of each day.

2 What we cover

Unit	What you do	Share of year
Robotics and control	Program a TPBot with the BBC micro:bit (MakeCode blocks, then MicroPython) through a full design cycle	Most
Computational thinking	Decomposition, pattern-spotting, algorithms, and Bebras challenges	A slice
AI and typing	What AI tools actually do, how to use them honestly, and touch-typing	End of year

3 What you need

- **Laptop** — charged. This is a practical subject and a flat battery is a lost lesson
- **Ring binder** — to keep worksheets, handouts, and practice paper
- **Dividers** — for the ring binder, to keep units separate
- **Writing utensils** — pencils, pens (multiple colours preferred), highlighters (multiple colours preferred), a ruler, and an eraser
- Your planner



A ring binder with dividers, like this one

4 How you are assessed

You are assessed against the four MYP Design criteria, each marked out of 8. The robotics unit runs through all four — that is what makes it a proper design-cycle project:

Criterion	What it looks like here
A Inquiring and analysing	Working out what the control problem really is, and researching how others solved it
B Developing ideas	A design specification, and designs you can actually justify against it
C Creating the solution	Building and programming the robot, with a record of how you got there
D Evaluating	Testing against your specification and saying honestly what you would change

Your grade comes from your **most recent, most consistent** work against each criterion, not an average of everything. Early attempts are for learning; they do not follow you around.

Criterion C is not "did the robot work". A robot that fails, where you can explain precisely why, scores better than one that works and you cannot.

5 Homework

Little, and not weekly — with one lesson a week the work belongs in the room. When it is set it is usually finishing a piece of design-cycle writing you started in class, and I will give you a week for it.

6 Working with the kit

The robots are shared, expensive, and easy to break by accident.

- Take the kit you are given and put back the kit you were given — same box, same cables, counted.
- Report anything broken or missing at the end of the lesson. You are not in trouble for reporting it; you are in trouble for leaving it for the next class to find.
- Do not reprogram, borrow from, or "fix" another group's robot.
- Nothing goes on the floor, and nothing drives off the desk.
- Save your code before the end of every lesson. Unsaved work is lost work, and with one lesson a week that hurts.

7 Deadlines and late work

Deadlines are given in class and posted on the course page. Work handed in late is marked and returned, but recorded as late; more than a week late is recorded as not submitted.

If something is going wrong — illness, deadlines stacking up, anything at home — **come and tell me before the deadline**. An extension asked for in advance is nearly always fine. Asking afterwards is different — by then, I usually cannot say yes.

8 Absence

Being away does not remove the work. It only delays it. **It is your job to find out what you missed** — check the course page first, then ask me.

- **Lessons.** Notes and resources are on the course page. Catch up before the next lesson.
- **Assessments.** Take the assessment within three school days of coming back. Come and book a time yourself — this is your responsibility, and I will not remind you.
- **Extended absence.** See me on your first day back and we will agree a realistic plan and a set of dates.

9 Phones, laptops, and AI

Phones go straight in your locker when you arrive. They stay there for the whole school day, in every lesson.

Laptops are open when the task needs them and closed when it does not. I will say which.

AI tools are genuinely useful and you should learn to use them well. In this course:

- **Fine:** asking it to explain something you are stuck on, generating extra practice questions, checking your own reasoning after you have done the work yourself.
- **Not fine:** getting it to produce an answer, a paragraph, a method or code that you then hand in as your thinking.
- **Always:** write one line at the end saying what you used and what for. Declaring it is never penalised. Not declaring it is a problem.

The reason is simple. The assessment has no AI in it. The moment you actually need to understand something has no AI in it either. If an AI does the work for you, you lose the time you needed to get better at design.

10 Academic honesty

You may work together — you learn faster when you do. But **the writing must be yours, done on your own.**

- Write down who you worked with, on the work itself.
- Cite anything you used — book, website, video, diagram, dataset. Putting it in your own words still needs a citation: you are borrowing the idea even when you change the words.

11 In the classroom

- Arrive on time and ready to start.
- Ask the question. Someone else has it too, and I would rather stop the lesson than let you fall behind.
- Wrong answers said out loud are how the lesson works. Nobody is laughed at in here.
- Leave the room ready for the next class.

12 How to do well in this course

Write things down as you go. The single biggest difference between a strong Design grade and a weak one is not the build — it is whether you kept a record of your thinking while you were

doing it. Nobody can reconstruct that convincingly three weeks later, and the ones who try always sound like they are guessing.

Try things. Break things (within reason). Then tell me exactly what happened.