# Preliminary instructions for starting with MetalUp.Express

1. Follow the ‘invitation link’ sent to you to see:

Text

Description automatically generated with medium confidence

1. **Log in** with one of the offered credential providers:

Graphical user interface, application

Description automatically generated

**Important:** please always log in using the *same credential provider* – otherwise you won’t be recognised.

1. You should now see the Home screen with some menus e.g.:

Chart, text

Description automatically generated

If you see a different screen please close the tab, go to <https://express.metal.org> and log in again – *and report the problem to me.*

You are now looking at the ‘Administration system’ (also known as the ‘Dashboard’). The menus, and actions within them, are determined by your role. Students see only actions to view their Assignments and their own User object; Teachers see the menus shown above; Authors (of projects) see additional actions.

1. Follow **Projects > All Assignable Projects** either selecting your preferred (programming) Language or leaving it empty to see all. There are currently four Projects available on the system in each of C#, VB, Python (no Java content as yet):

A picture containing table

Description automatically generated

1. Right-click on any Project to view its details, and then open its Actions menu e.g.:

A picture containing diagram

Description automatically generated

Note that when clicking on an object reference you may also *right-click* to open the view in the *other* half of the screen.

1. There are six (white) icons at the bottom of the screen. Hovering on any will explain its purpose. Try them all out.

A picture containing graphical user interface

Description automatically generated

Then get back to the view of the project, either by using the Back icon, or by retrieving it again from the Home view.

1. On the Project’s actions there there are also actions to **Assign To Indivdual** (typically a Student, but it could be to a Teacher colleague) and **Assign To Group** – but you can’t use them yet because you don’t have any Students or Groups (of Students set up yet). Instead, invoke **Assign To Me**. This returns a new Assignment object as shown below:

A picture containing graphical user interface

Description automatically generated

You can also retrieve this assignment via **Home > Assignments > My Assignments** – that’s how Students will view their assignments.

1. Click on the link **Start or continue Project**. This takes you out of the dashboard, and into the ‘Task View’ which is where the Tasks that make up the project are viewed and undertaken:

Graphical user interface, application

Description automatically generated

There are six panes, the relative vertical size of these panes may be adjusted using the handles in the bottom right corner of each pane, and horizontally using the handles on the dark background at the bottom-right of each column as highighted above. The six panels are:

**Task Description**

Explains what you need to do. Most projects have multiple tasks. The **Next Task** button is not enabled until the current task is completed. The **< >** buttons allow you to review tasks you have already completed, and return to the current one (they are not enabled here, because this is the first task in the project).

**Code**

This is where you write code – typically functions. As soon as any code is entered (or a previous version edited) the **Submit** button is enabled – the latter with either produce the message ‘Compiled OK’. Or will show errors. These errors may be the usual compilation errors you may get from the language selected, or it might be that your code violates rules enforced by the Task. In particular, the initial range of projects are all concerned with teaching a specific style of ‘functional’ coding where functions take a very simple form. This form is shown as a ‘prompt’ before you enter your own code e.g. (for Python):

Shape, rectangle

Description automatically generated

The **< >** buttons allow you to review previous versions of your code for this task. If an edit has not compiled, you may, this way, go back to a previous version, edit it, and **Submit** again.

**Test**

Most (but not all) Tasks define automated Tests and, where they do, the Task is not completed until all the tests have been run and passed. **Run Tests**however is only enabled when you have submitted new code in the **Code** pane, and it has successfully compiled.

You will either get back the message ‘All Tests Passed’, in which case the Next Task button on the **Task Description** will now be enabled.

**Hints**

Each task *may* offer Hints – commonly three, but could be any number. Taking a hint costs you potential Marks, and you will see these deducted from the **Marks Available** shown in the title of the **Task Description** pane. The **< >** buttons allow you to review hints you have already taken and return to the last one taken – reviewing previous hints and returning does not cost you additional marks.

Note that the *last* hint provides the user with the *full code* required to pass the tests. However, the user must still manually re-type this code into the **Code** pane, successfully submit (compile) it, and run the tests.

**Expression & Result**

Here you may enter and evaluate any expression written in the syntax of the selected programming language. This may be as simple as:

3 + 4 [enter]

The result is shown in the **Result** pane underneath.

Your expressions may include function calls, including to functions you have written (and successfully compiled) in the **Code** pane and/or other ready-written functions that are made available to you for this Task. You may use the **Expression** pane at any time, to test your functions manually to see if they are returning the result expected forgiven arguments.

Sometimes the **Task Description** will request or suggest that you try out specific expressions as part of the task.

**Completion**

When you have completed the last Task in a project, you will see an additional button, **Return to Assignment**. This takes you back to the Assignment object in the Dashboard view, where you will see that the Status and the Marks have been updated.

Note that you may leave the Task View (or the whole application) at any stage *without losing your work****.*** You can return to MetalUp.Express find the Assignment and click the **Start or continue Project** link again, and which point you will be able to navigate to the last Task you were working on, with the last *submitted* code shown in the **Code** pane, and with the Marks as they were (sorry – it also remembers which Hints you have taken!)

This is because all significant ‘activites’ (submit code, run tests, take hint) are recorded as Activity objects associated with the Assignment. You can view these via **Actions > List Acitivities** on the Assignment object.

When you come to assigning projects to your Students you will find these Activities very useful. For example you can see how much time elapsed and/or how many code submissions were attempted before taking the next hint - useful to identify students that aren’t really trying.

There are many other features of the Administration system not covered in this introduction: how to invite Students (or Teacher colleagues) to register; how to create Groups of students (not just formal sets, but possible just groups of students within a set with specific ability or difficulty. How to track progress of Assignments.

You can explore some of the actions for yourself, but please remember that this is currently *a system under development*. Meantime, it is quite possible that there are still bugs in some actions, so…

**Please report any bugs or issues – in the Admin system or in the Project contents (Task Description, Hints, Tests etc) by emailing me with full details, screenshots if possible. F**eel free to ask for help if anything isn’t clear. And let me know how you get on.

Richard Pawson

[rpawson@metalup.org](mailto:rpawson@metalup.org) (or any other email you have for me)

8th December 2022.