



COSC1073 - Programming 1

TuteLab 2: Decisions, Repetition, Parameters and Arrays

1. Complete the following Java code segment so that the messages “You have passed COSC1073” and “You may proceed to next level” will be printed (using two separate println statements) if the input marks is greater than or equal to 50. Otherwise, the messages “Sorry, you have failed COSC1073” and “You may re-enroll next semester” should be displayed (using two separate println() statements).

```
// assume student marks is read and stored in the variable marks
if ( ... )
...
```

2. Write a program to compute the overall result for the subject CC123. The subject has a course hurdle and an exam hurdle both of which are marked out of 100. A student is considered to have passed the subject if the mark for both each of the hurdles are greater than or equal to 40 AND the overall mark (the average of two hurdles), is greater than or equal to 50. All other cases are considered to be a failure.

The formula for the overall mark is: $(\text{course result} + \text{exam result}) / 2$

Test your program with the following data. The expected result is shown next to it.

```
course = 45 exam = 65 (expected result is pass)
course = 30 exam = 90 (expected result is fail)
course = 30 exam = 36 (expected result is fail)
course = 60 exam = 90 (expected result is pass)
```

Start by checking for the “pass” criteria in a single if statement first and handling the “failure” case in an else clause.

3. Write for loops / nested for loops to print the following.

- a) 2 3 ... 10
- b) 20 30 40 ... 100
- c) 1 2 4 5 7 8 10 11 ... 98 100 (numbers from 1 to 100 which are not a multiple of 3)
- d) 1 2 3 ... 10
2 4 6 ... 20
.
.
.
10 20 30 ... 100
- e) 1 2 3 ... 10
11 12 13 ... 20
.
.
.
91 92 93 ... 100
- f) a b c ... z
- g) A B C ... Z

4. Write code segments to produce the following output.

a)	12345	b)	5	c)	*	d)	*
	1234		54		**		***
	123		543		***		*****
	12		5432		****		*****
	1		54321		*****		*****

5. What will be the output of the program below?

```
public class TestArray2 {
    public static void main (String[] args) {
        int nums1[] = { 10, 25, 20, 40, 10};
        int nums2[] = new int[5];
        for (int i = 0; i < 5; i++)
            nums2[i] = nums1[4-i];
        for (int i = 0; i < 5; i++)
            System.out.printf("%d ", nums2[i]);
        System.out.println();
    }
}
```

6. What will be the output of the program below?

```
public class TestArray3 {
    public static void main (String[] args) {
        int nums1[] = { 10, 25, 20, 40, 10};
        int nums2[] = { 30, 50, 67 };
        // size of new int array = size of 1st array + size of 2nd
        // array
        int nums3[] = new int[nums1.length + nums2.length];
        for (int i=0 ; i< nums1.length ; i++)
            nums3[i] = nums1[i];
        int len = nums1.length; // length of first array
        for (int i=0; i<nums2.length; i++)
            // copying from 2nd array but with offset = len
            nums3[len+i] = nums2[i];
        for (int i=0 ; i< nums3.length ; i++)
            System.out.println(nums3[i]);
    }
}
```

7. Write a code segment to find the average of all the values in the array.

```
public class FindArrayAverage {
    public static void main (String[] args){
        // array of random size filled with random values
        int [] values = { ... };
        // your code starts here
        // ...
    }
}
```

8. Write a code segment to find the largest value in the array.

```
public class FindArrayLargest {  
    public static void main (String[] args) {  
        // array of random size filled with random values  
        int [] values = { ... };  
        // your code starts here  
        // ...  
    }  
}
```

Additional Exercise: GitHub using the rmit-p1-s1-2021 Organisation → next page

GitHub using the **rmit-p1-s1-2021** Organisation

Prerequisite

It is assumed you have completed the GitHub material found in the TuteLab 1 sheet.

GitHub **rmit-p1-s1-2021** Organisation

All repositories for this course **must** be hosted within the **rmit-p1-s1-2021** organisation. You should create a GitHub account using your RMIT student email address; the username for the GitHub account should ideally be your RMIT student ID, for example **s1234567**. If that username is not available you can append additional information, for example your name **s1234567-matthew**, etc...

You will be emailed an invitation to join the **rmit-p1-s1-2021** organisation to your RMIT student email address. If you have not received the invitation or encounter any issues, please send an email to:

matthew.bolger@rmit.edu.au

Ensure GitHub Desktop is correctly configured

The following material assumes you have joined the **rmit-p1-s1-2021** organisation and are signed into your GitHub account in GitHub Desktop.

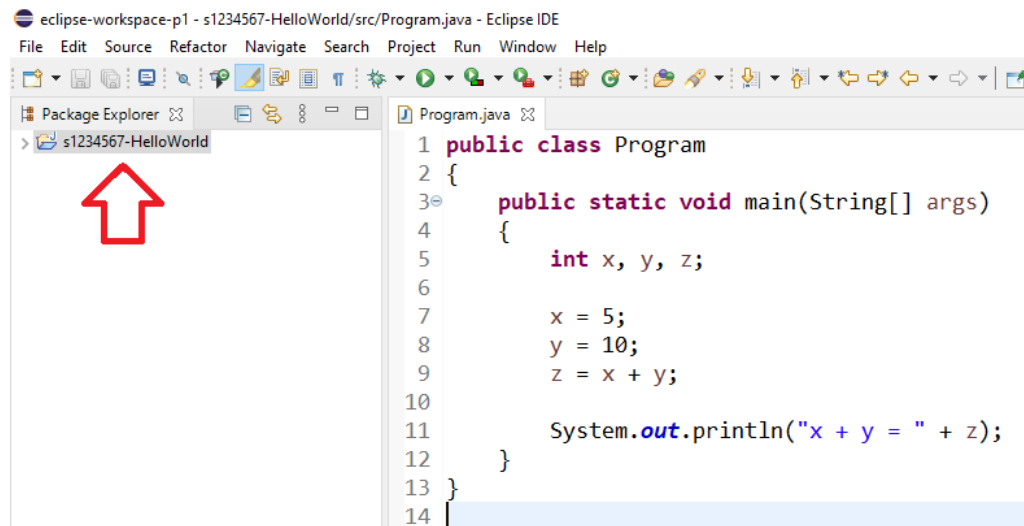
To ensure you are logged into GitHub Desktop with the correct account, select File -> Options:

The image shows two screenshots of the GitHub Desktop 'Options' dialog box. The top screenshot shows the 'Accounts' tab selected in the left sidebar. Under 'GitHub.com', there is an account for 'matthewbolger' with the email '@matthewbolger'. A red arrow points to this account. There is a 'Sign out' button next to it. Under 'GitHub Enterprise', there is a 'Sign in' button. The bottom screenshot shows the 'Git' tab selected. It has fields for 'Name' (containing 'Matthew Bolger') and 'Email' (containing 'matthew.bolger@rmit.edu.au'). Red arrows point to both of these fields. Below these fields is a section for 'Default branch for new repositories' with radio buttons for 'main' (selected), 'master', and 'Other...'. At the bottom of the dialog are 'Save' and 'Cancel' buttons.

Publish Repository to GitHub **rmit-p1-s1-2021** Organisation

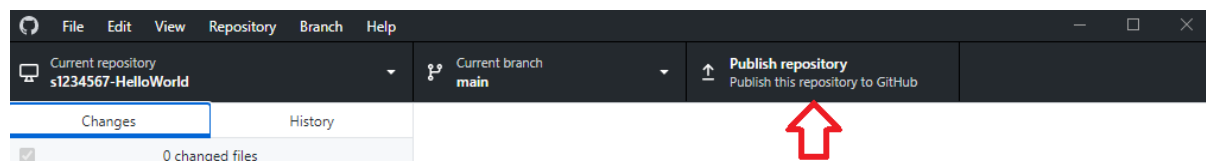
When publishing a repository to the **rmit-p1-s1-2021** organisation the repository name must be unique, thus it is ideal that you prefix all your repositories with your RMIT student ID to avoid naming conflicts with repositories already present within the organisation.

In this example the project has been called **s1234567-HelloWorld**; in TuteLab 1 the project was called **HelloWorld**.

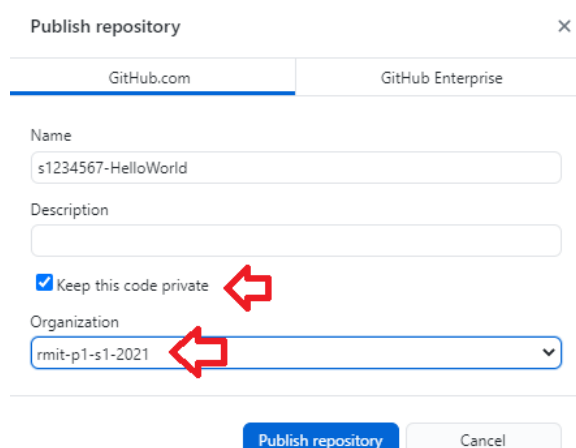


A local repository for this project has been created; the steps to do this are present in TuteLab 1.

In GitHub Desktop click the **Publish repository** button:

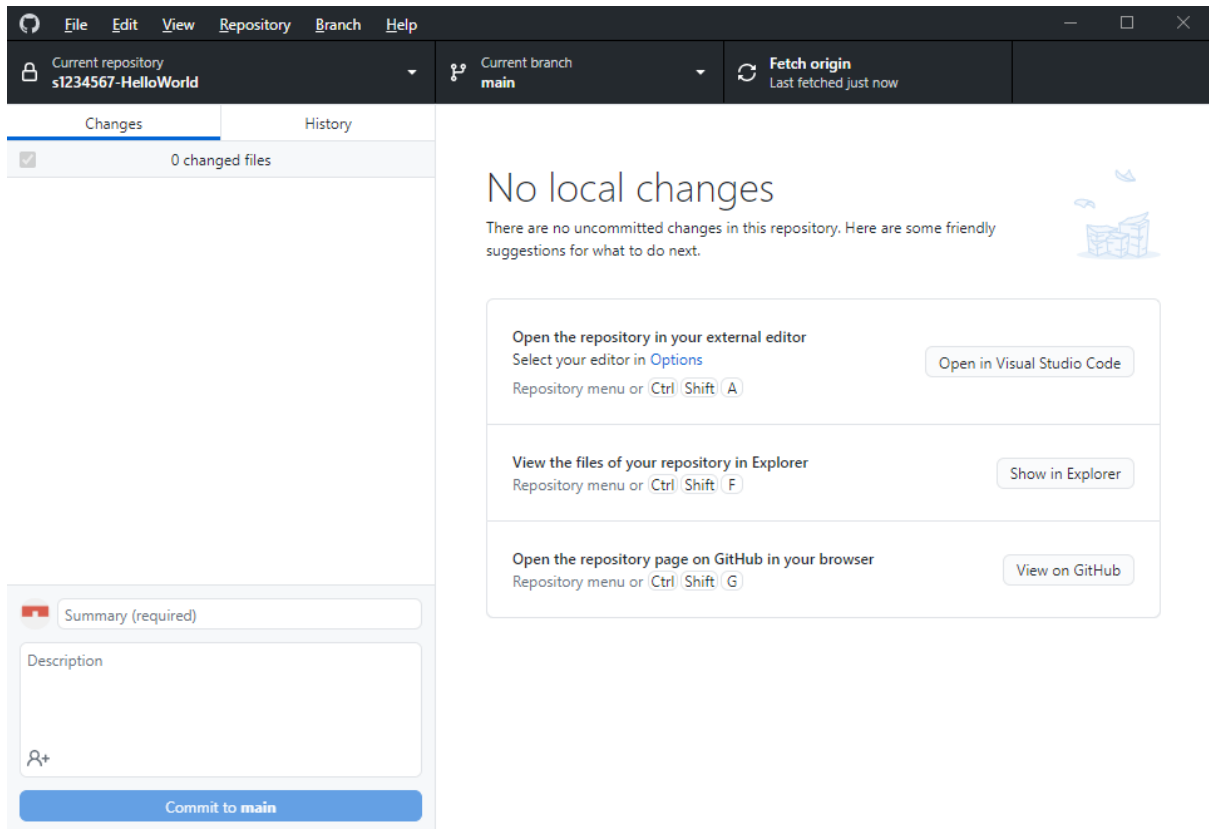


Then ensure **Keep this code private** is checked and set the Organisation to **rmit-p1-s1-2021**:



Then click the **Publish repository** button.

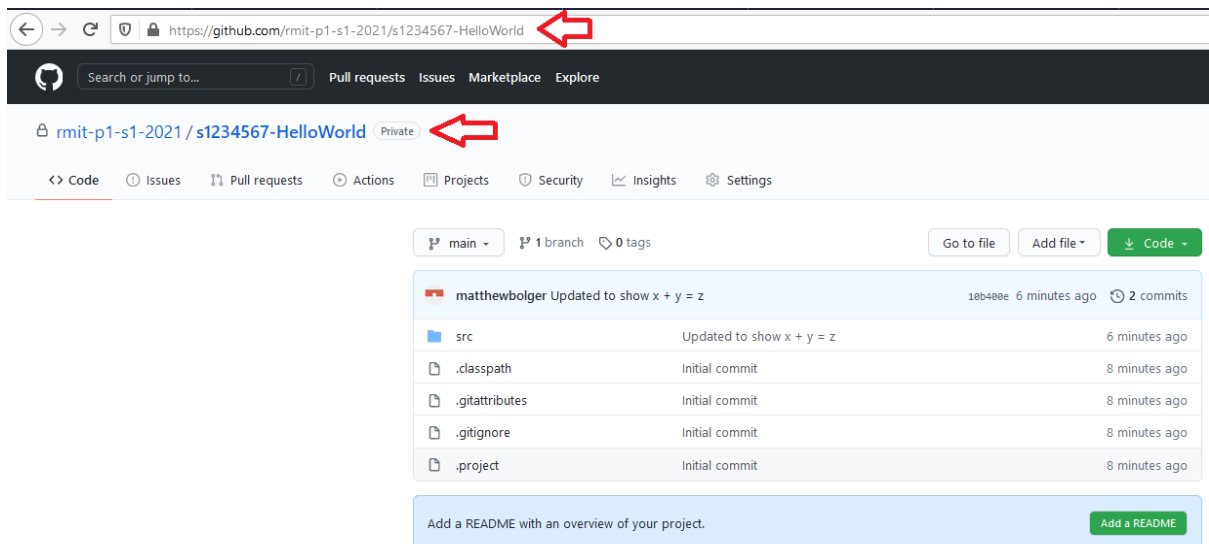
GitHub Desktop should now show the following:



Note: The **Publish repository** button is now replaced with **Fetch origin**.

You can also check the GitHub repository in your browser, the URL should include the **rmit-p1-s1-2021** organisation; for this repository, the URL is:

<https://github.com/rmit-p1-s1-2021/s1234567-HelloWorld>



Note: The repository is within the **rmit-p1-s1-2021** organisation and is **private**.

Committing and Pushing

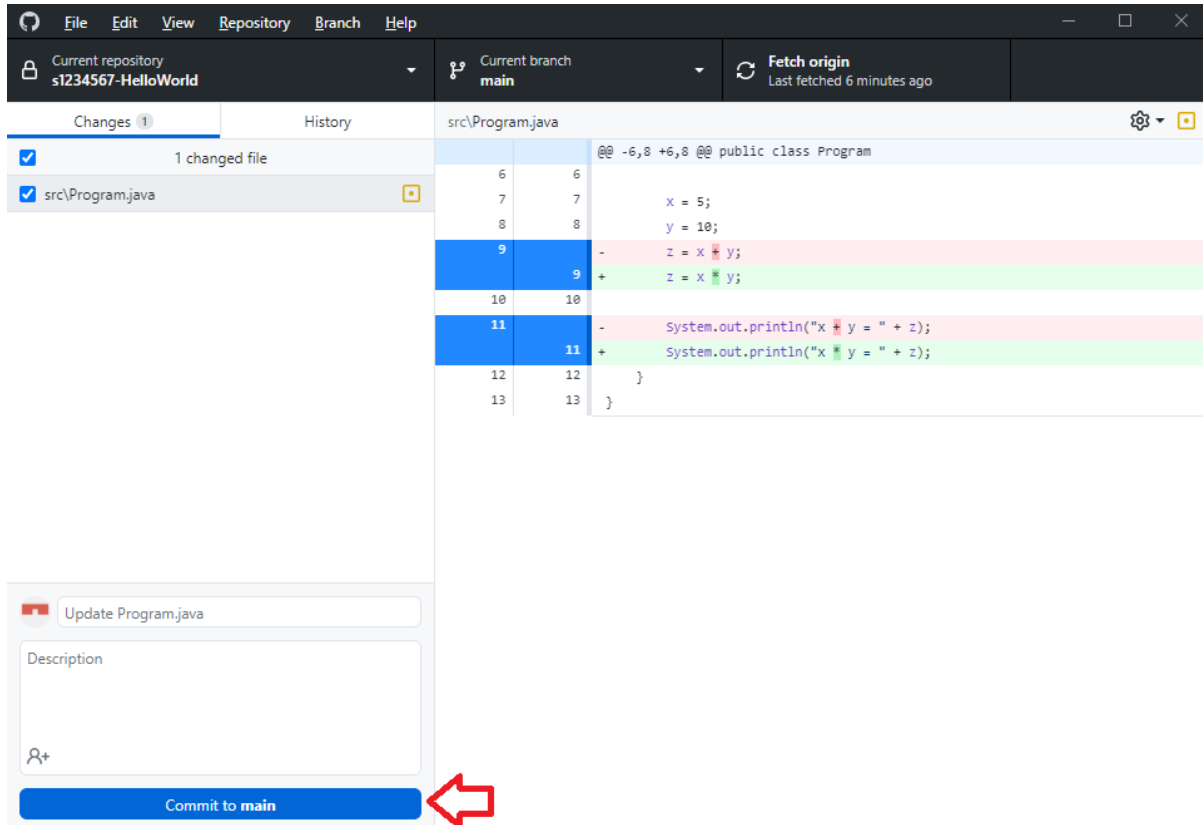
When a commit is made to the repository, you will now have the option to push your local commits up to the online repository.

Consider the following change:

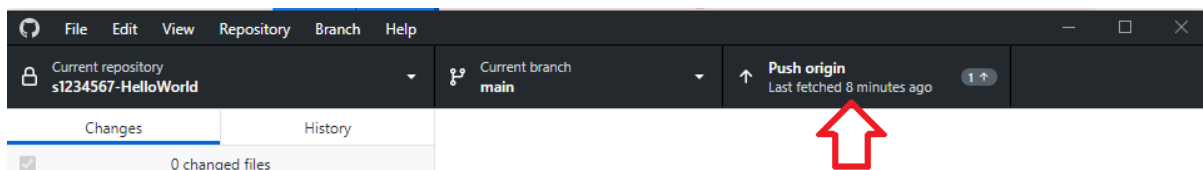
```
z = x * y;
```

```
System.out.println("x * y = " + z);
```

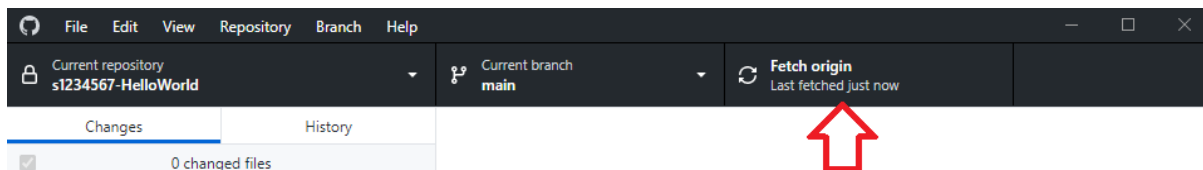
Now z is equal to x * y and the output is changed from + to *. After making this change in Eclipse, GitHub Desktop detects this change. You can now commit this change:



After committing the **Fetch origin** option will change to **Push origin**:



Clicking **Push origin** will push the commit to GitHub. After pushing the **Fetch origin** option will return:



The commit can also be viewed on GitHub; refresh the page in your browser:

The screenshot shows a GitHub repository page for 'rmit-p1-s1-2021/s1234567-HelloWorld'. The repository is private. The main branch is 'main' with 1 branch and 0 tags. A commit by 'matthewbolger' titled 'Update Program.java' is highlighted, with a red arrow pointing to it. The commit hash is '3e4bf19' and it was made '3 minutes ago'. A red box highlights the '3 commits' link. Below the commit list, there is a table of files:

File	Commit	Time
src	Update Program.java	3 minutes ago
.classpath	Initial commit	16 minutes ago
.gitattributes	Initial commit	16 minutes ago
.gitignore	Initial commit	16 minutes ago
.project	Initial commit	16 minutes ago

At the bottom, there is a button to 'Add a README with an overview of your project.'