# Computer Sciences: Programming (CIS1048-N) Worksheet 6: Design, Testing and Version Control

## Before you start

Watch the video lecture content (when available) before attempting these exercises.

Ensure you have attempted all tasks from Worksheet 5. Any issues seek help from your tutors.

Attempt to complete this set of tasks before your next session. Reminder, questions that start with **[ASSESSMENT]** will be submitted at the end of the module to provide you with an overall grade.

## Introduction

This session aims to familiarise you with design (using UML flow diagrams), testing your programs, and using the Git version control system with GitHub. This week is comparatively light on code, but **heavy on screenshotting**!

**Hint:** On Windows, remember that GitHub Desktop will take a lot of the learning curve out of using Git/GitHub. On Linux, things get slightly more complex. Get in touch for more information on this if you would like it!

**Questions begin on the next page.**

## **[ASSESSMENT]** Question 1: Flow Diagram to Code (flow\_to\_code.py)

Let’s start by implementing some code from a flow diagram. Below, you will see a UML flow diagram:

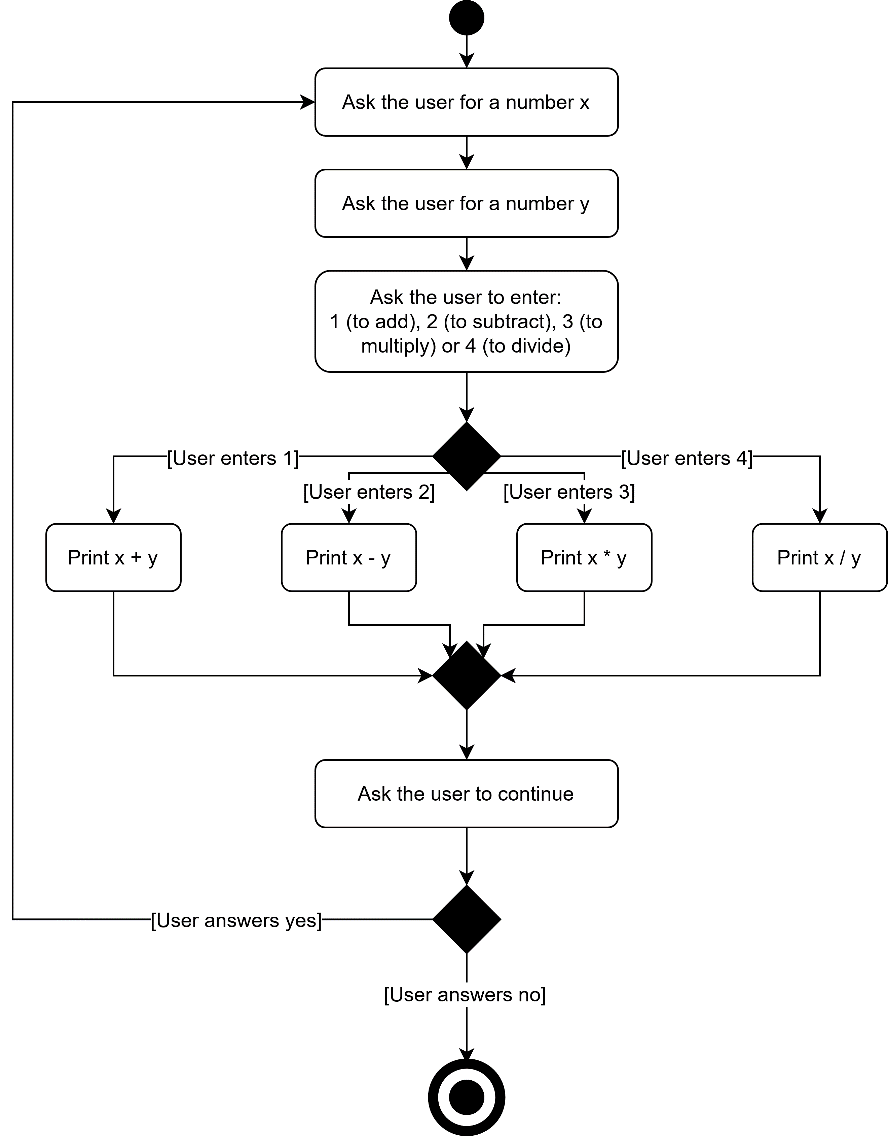


Figure 1: UML flow diagram

1. Turn this diagram into a Python program.
2. Ensure your program implements this functionality exactly with thorough manual testing.

## **[ASSESSMENT]** Question 2: Code to Flow Diagram (high\_low\_flow.io)

Under the week 6 worksheet materials on Blackboard, you will find a Python file titled high\_low.py.

1. Download this file and play with it, inspecting its code. Take the time to understand how it works.
2. Using the notation shown in the lecture, reverse-engineer this program into a flow diagram. **You do not have to go into statement-level detail!** Try to make your flow diagram abstract, like the one given in question 1 on this worksheet. It is recommended to use the following free online tool to create your flow diagrams: <https://draw.io/>

## **[ASSESSMENT]** Question 3:Creating a GitHub Account (github\_1.png)

Now, we’re going to create a GitHub account and commit our code to it. **This question involves taking screenshots for submission, please follow it carefully.**

**Note:** Many of you have done this already, in which case you should still do this question please.

1. Go to <https://github.com/> and sign up for an account.
2. Also, optionally, go to <https://education.github.com/pack> and add your academic e-mail address (@tees.ac.uk) to get the GitHub student developer pack (it’s pretty cool!) Enter “coursework” or similar as your reason for requesting this pack.
3. Document your GitHub account for the assessment by taking a screenshot of your GitHub profile page (save as **github\_1.png**).

## **[ASSESSMENT]** Question 4: Creating a new Repository (github\_2.png)

Now, we’re going to create a new public GitHub repository for us to clone using GitHub Desktop. This question involves taking screenshots for submission, please follow it carefully.

1. Go to the GitHub website and create a new **public** repository and tick the “Initialize this repository with a README.md” checkbox. Make sure the repository is called *exactly* this: python-hello-world
2. From here, take a screenshot of the page of the new, mostly empty repository (save as **github\_2.png**).

## **[ASSESSMENT]** Question 5: Using GitHub Desktop (github\_3.png to github\_6.png)

We’ll now download the GitHub Desktop GUI client and get started with it.

**Note:** This question involves taking screenshots for submission, please follow it carefully.

1. Download the version of GitHub Desktop for your operating system from: <https://desktop.github.com/>
2. Get it set up by logging into it using your GitHub account credentials just like you would through the website. Once this is done, screenshot the main GitHub Desktop window and save as **github\_3.png**.
3. Now, clone down your python-hello-world repository. Take a screenshot of what the window looks like now and save as **github\_4.png**.
4. Add a new file called hello.py and add print("Hello world!") to it. Take a screenshot of what the GitHub Desktop window looks like now and save as **github\_5.png**.
5. Now, commit changes and push. Open the “History” tab in the GitHub Desktop software, screenshot this and save as **github\_6.png**.

## Document History

Revision 0 (21-Feb-21): This is the initial version of the 2020/21 exercise.