An MHRD Govt of India Initiative

Virtual Lab Development Guide :

by Virtual Labs IIT Bombay.

vlabs.iitb.ac.in

Table of Contents

Virtual Labs

Vlabs is a project initiated by the Ministry of Human Resource Development, Government of India, under the National Mission on Education through Information and Communication Technology. The project aims to provide remote-access to Laboratories in various disciplines of science and engineering for students at all levels from undergraduate to research.

Objectives of the Virtual Lab

- To provide remote-access to Labs in various disciplines of Science and Engineering. These Virtual Labs would cater to students at the undergraduate level, post graduate level as well as to research scholars.
- To enthuse students to conduct experiments by arousing their curiosity. This would help them in learning basic and advanced concepts through remote experimentation.
- To provide a complete Learning Management System around the Virtual Labs where the students can avail the various tools for learning, including additional web-resources, video-lectures, animated demonstrations and self-evaluation.

- To share costly equipment and resources, which are otherwise available to a limited number of users due to constraints on time and geographical distances.

About the Boot Camp

The NMEICT Virtual Labs project has created ~117 labs (1 Lab is a collection of 8-10 experiments) covering concepts in various engineering and science curricula. These can be accessed at http://vlabs.ac.in.

However, there remains a need for additional content to be created, primarily to fill gaps in syllabi, but also towards developing improved versions of content already hosted. Consequently, we encourage Lab Developers to come forward to develop such content. Boot Camp aims to facilitate the creation of such content.

Participants will be creating new experiments and their creation shall be hosted live during the Boot Camp on the Community Portal Vlabs-Dev; we will arrange for expert review of content created.

Objectives of Boot Camp

- The aim of the bootcamp is to develop a collaborative community of Vlab developers across a wide spectrum of disciplines and locations, such that Virtual Labs has content to reflect local curricula in all parts of the country, and remains open to the possibility of change.
- The Boot camp for Vlabs Development Certification is a short, intensive, and rigorous course for accelerated training. Its aim is to consolidate the learning process into a short, productive capsule for motivated learners and career professionals.
- The boot camp will begin at 8AM each day and go on till 8 PM, with breaks for lunch and refreshments. Participants will learn key concepts for the development of Virtual Labs content, and have hands-on training sessions for the same. Each successful participant developing an experiment will be given a certificate indicating their achievement.

Chapter 1: Pedagogy of Virtual Labs

Introduction and Overview

Virtual Lab is a collection of many virtual lab experiments. Each experiment would cover certain aspects of a lab. Designing a Virtual Lab experiment consists of well defined logical steps. These steps include

1. **Round 0** - It consists of a brief overview of the virtual LAB and should contain broad objectives, outcomes, brief overview on software used etc. One has to also mention the experiments that will be covered in this LAB.

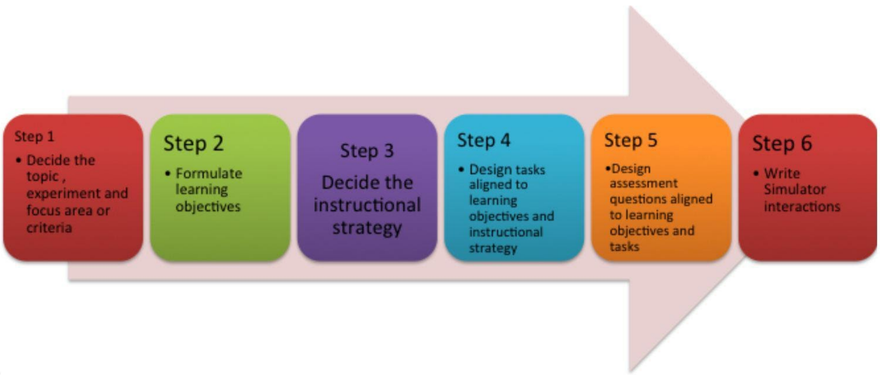
2. **Round 1 (Pedagogy)**- From this stage, every experiment in a given lab needs to have a seperate documentation. In this stage learning objectives and pedagogy for experiments needs to be clearly defined

3. **Round 2 (Storyboard)**- After freezing the pedagogy and the instructional strategy. One can start designing the story, mindmap, flowchart and storyboard in the section

4. **Round 3 (Lab Manual)**- In round 3, the developer designs the lab manual and the programmers code for the virtual lab simulator At this point the developer should form groups with other faculty and students and begin the coding for the experiment simulation. The coding should be done ONLY in FOSS (Free and Open Source) based software

5. **Round 4 (Code Development)**- In this stage integration of different components should be done using the LDK (Framework-Template). And when the lab template is ready, it must be submitted for approval.

6. **Round 5**- This is the final stage where review and approval would be done by the virtual labs team.



1.1 Focus Area

Focus area is the area of focus in any virtual lab experiment. There are few main focus areas:

1. **Reinforce theoretical concept:** Reinforcing a concept that is required for a student to understand higher level concepts is very important. One must define this focus area if the developer wants to reinforce previously learnt concept.

2. **Instrumental and Practical Skills:** This area focuses on instrumentation, practical skills and demo experiments. In other words, the student is required to acquire the skill of using an instrument or learn the process of an experiment rather than focusing on data analysis.

3. **Experimentation:** Experimentation focuses on the process of experimentation including the protocol and understanding different methodologies in an experiment.

4. **Data Analysis:** Data analysis as the name suggests involves analyzing data obtained after performing the experiment. Some experiments may have rigorous data analysis and conclusions are based on the data analysed.

5. **Modelling**: There are experiments in which a student might have to apply different models or different theories to understand the working of an experiment or the theory behind the experiment.

6. **Learn from failure:** Learning from failure is one of the most important areas where students can take full advantage of virtual lab technology. One can understand how malfunctioning of an instrument can result in error or failure and can repeat the experiment correctly in virtual environment.

While choosing these focus areas, one can choose multiple focus areas for a given experiment. But one has to keep in mind that the focus areas chosen should be well justified and logically selected to avoid any discrepancy with learning objectives and instructional strategy.

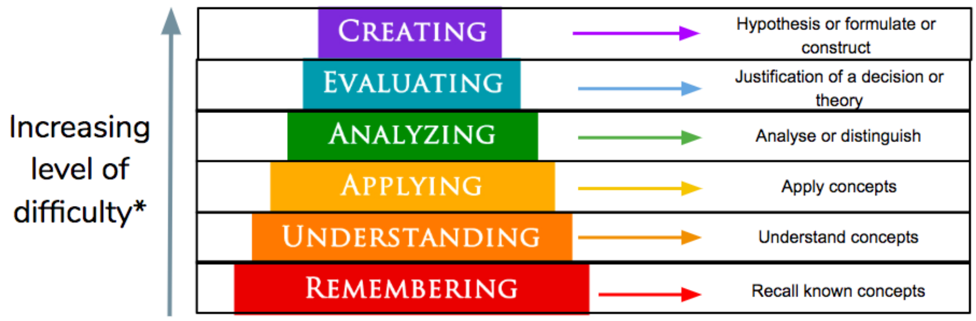
1.2 Learning Objectives and Bloom's taxonomy

1.2.1 Learning Objectives

Learning objectives are objectives set by the proposer or instructor that has to be achieved. Learning objectives help in defining the accurate coverage and flow of a virtual lab simulation. Defining proper learning objectives acts as a checkpoint for a developer to ensure complete coverage of a virtual lab experiment.

1.2.2 Bloom's Taxonomy

But in order to write good learning objectives Bloom's taxonomy should be used. Dr Benjamin Bloom invented a way of classifying learning objectives now known as Bloom's Taxonomy. There are 6 levels of objectives or cognitive levels in Bloom's taxonomy which should be used in developing an experiment This taxonomy classifies learning based on cognitive skills.



*need not be true for all, weak students may find remembering and understanding difficult too.

Cognition means, the process of acquiring knowledge through means like thinking, reasoning, observing, memory recall etc. Some of the cognitive skills are easy for students to grasp, but some are very difficult. Bloom's classified the important cognitive skills to improve learning and promote learning skills in students.

1.2.3 Bloom's Taxonomy: Action Verbs

Further these cognitive levels are mapped to action verbs. Action verbs are words that define some action. Examples of action verbs include describe, explain, state, apply, solve etc. Bloom's taxonomy action verbs are only "single word that carry an action". These action words describe the cognitive processes by which thinkers encounter knowledge.

A complete set of verbs which can be used to formulate learning objectives at the various levels of Bloom's taxonomy are given in the following table.

Level	**Action Verbs**
Create	Adapt, Build, Change, Choose, Combine, Compile, Compose, Construct, Create, Delete, Design, Develop, Discuss, Elaborate, Estimate, Formulate, Imagine, Invent, Make up, Maximize, Minimize, Modify, Plan, Predict , Propose, Solve, Suppose, Test, Theorize
Evaluate	Agree, Assess, Choose, Compare, Conclude, Criticize, Decide, Deduct, Defend, Determine, Disprove, Estimate, Evaluate, Explain, Interpret, Judge, Justify, Mark, Perceive, Prioritize, Prove, Rate, Recommend, Rule on, Select, Support
Analyse	Conclude, Infer, Compare, Analyze, make an assumption, Categorize, Classify, Contrast, Discover, Dissect, Distinguish, Divide, Examine, Inspect, List, Relate, Simplify, Take part in, Test for
Apply	Select, Apply, Calculate, Build, sketch, Choose, Construct, Develop, Experiment with, Identify, Interview, Make use of, Model, Organize, Plan, Solve, Utilize
Understand	Describe, Explain, Give example of, Select, Demonstrate, Classify, Compare, Contrast, Extend, Illustrate, Infer, Interpret, Outline, Relate, Rephrase, Show, Summarize, Translate
Remember	Describe, Select, Show, Define, Find, Label, List, Match, Name, Recall, Relate, Spell,Choose, Tell

While writing learning objective one should ensure the sentence starts with an action verb. For example, an LO could be:

1. Describe how a cloud formation occurs during formation of hurricane
2. Explain different factors that affect friction
3. List the countries which had low GDP than India in the year 2017-2018

So in all the examples above the statement has begun with an action verb followed by the instruction or objective. It is also important to use the right action verb to achieve the correct LO.

Learning objectives –Do's & Don'ts:

As per the definition of learning objectives they need to be very specific and measurable hence you should not use verbs such as understand, visualize etc instead use action verbs such as identify, list, describe, solve etc.

Similarly the learning objective should be concerned with the learner and not the teacher hence avoids using verbs such as teach, show, demonstrate etc.

The learning objectives should be clearly listed down and a cognitive level should be assigned to it. For example:

Sr.No.	Learning Objective	Cognitive Level
1	Identify the set-up, test section, pumping mechanism and measurement of flow rate and pressure drop	Recall
2	To enable the student to describe the methodology to perform the experiment	Understand
3	Use the fundamental and empirical equations to calculate the various entities	Apply
4	Examine and compare the calculated values with the experimental ones	Analyze
5	Realize the significance of controlling variables/parameters and visualize their effect on the desired entity	Evaluate
6	Solve an open-ended problem related to the existing infrastructure or real world	Create

It is expected to develop a lab or prepare experiment pedagogy, based on 6 cognitive levels namely; RECALL, UNDERSTAND, APPLY, ANALYZE, EVALUATE and CREATE. It is expected to lay more emphasis on the last three levels. However none or very few of the physical experiments touch upon the last two cognitive levels.

For e.g. in the case of the experiment in Fluid Mechanics, Friction in Circular pipe: the first step may be to include the first four cognitive levels.

Recall	Identify the layout of the experimental set-up, the equipment and the instruments used therein, various technical terms involved (Reynolds' number, friction factor, pressure drop etc.). This is based on the theory given in the manual/experiment.
Understand	The student should know and understand the rationale of the experiment and the procedure to perform it. This also includes knowing the functionality of the various devices used (rotameter, manometer, pump etc.), the flow patterns and the connectivity with the real world. Additionally the understanding the theoretical correlations is equally important.
Apply	Based on the experimental observations and the data generated/collected and the theoretical correlations, the student should be able to calculate the various entities (velocity, pressure drop and friction factor) and summarize the same using relevant plots.
Analyze	Based on the observed data and the calculations the student should be able to appreciate and compare the observed and the theoretical values and the variables/parameters affecting the same. In the traditional version this falls under Results and Discussion
Evaluate	Based on the obtained values and the plots, the student should be able to put forth reasonable conclusions. Further to this it should be possible to go one step beyond this and think about the effect of change in the architecture (diameter of pipe, length of pipe etc.) on the result.
Create	This aspect involves a segment of open-endedness. Students should be able to or motivated to think out of the box and propose alternate ways to perform the experiment. This may involve proposing different but viable analytical tools or even multiple test sections to study the effect of additional variables. For example in the case of Friction in Circular Pipe this level may involve proposing pressure sensors/transducers for pressure measurement and pipes of different sizes, cross-section and length to study their effect on the friction factor. However simulation for the same calls in for real time data which either needs to be generated or should be available from the literature. This calls in for a possible integration with the modular laboratory set-ups which can help address this aspect.

1.3 Instructional Strategy

Introduction

Instruction is a combination of teaching and learning activities. Instruction is a whole process that includes facilitating the learning process and guiding the students. Strategy means “plan of action”. Instructional strategy helps in choosing which approach or strategy to use while

Types of instructional strategies:

There are four types of instructional strategies:

Expository

‘Do as per the instructions’ - Stereotype protocol. Almost all the physical labs follow this methodology. Thus it addresses only the first three cognitive levels. Moreover, it does not involve critical thinking and analysis.

Guided inquiry

This includes the primary level of open-endedness. The objectives and the theory is explained or given. The students are asked to propose a methodology to address the objectives leading to data analysis, Results and Conclusions. For e.g. in the case of Friction in Circular Pipe the students may be given the test section and a pump. The students are supposed to find alternate but feasible ways to measure the flow rate and the pressure drop. They may use a graduated cylinder or a two-tank system to measure the flow rate. For measuring the pressure drop they may use a pressure gauge, pressure sensor or transducer instead of a manometer. Such strategies can be implemented with relative ease in Computer science, CAD, Electrical Engg. and to a certain extent in Physical sciences.

Problem based

This is based on posing a problem associated with the theory with the available lab resources. The learning objectives are often defined. The students are expected to brainstorm and propose a design methodology. This could be similar to constructing something from a Mechano set. For e.g. in the case of Friction Factor determination in a pipe, the student has to figure out the cross-section, diameter and the length of the pipe based on the available resources and then assemble them. This may also involve selecting and proposing the right type of pump. This protocol should ultimately lead to data analysis, comparison, results & discussion and conclusions.

Project based

This to a greater extent overlaps on the Problem based strategy with the only difference being the problem statement is associated with the real world. In the case of determination of friction factor, the problem may be posed as follows: ‘A pipeline needs to be designed for given flow conditions with a limitation on the pressure drop and the pump required to cater to this service be specified’. Thus, the objectives are well defined but the remaining protocol needs to be proposed by the students.

1.4 Task and Assessments

Assessments are tools an instructor can use to understand whether students have understood the concept. One way to assess students is to make the students perform a task in the simulator. While performing the task, students will have to use different skills like observation, analysis, logically predict outcome etc. and then infer the solution. A simple solution to deciding task is to check for learning objectives defined for that experiment and build task and assessment corresponding every learning object.

Assessment methods

Typically two types of assessment methods are used namely; Formative and Summative. Almost all the Institutes in India use the summative method wherein the assessment is done after performing the experiment or towards the end of the semester during the viva examination. Formative is often quite rigorous and time consuming. On the VLabs platform formative may mean that when the student enters a value which needs to be calculated and if the same in incorrect, the virtual simulator will display an error message which will eventually reduce the credits for a given experiment (this aspect needs to be incorporated while coding). Alternatively a question will be posed by the simulator prior to entering the value prompting the student to enter the range in which the value is likely to lie.

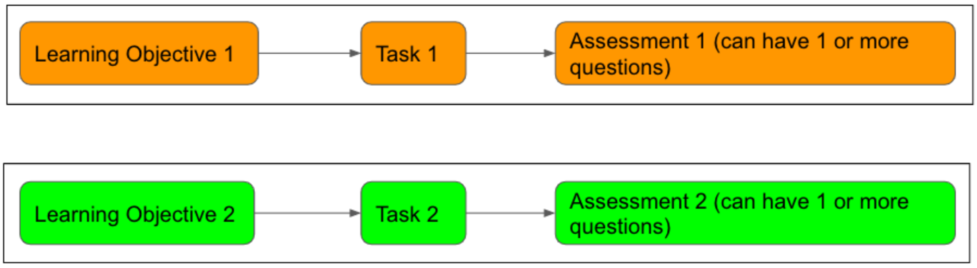
Formative assessment:Assessments part of the learning process to enhance or understand whether a student is able to grasp. This happens during the process of learning.

Summative assessment:Summative Assessments: Assessments conducted after the learning has been completed to test student’s understanding of a concept. This happens after the process of learning.

Relating LO’s, Assessments and Tasks

A simple solution to deciding task is to check for learning objectives defined for that experiment and build task and assessment corresponding every LO.

For example if an experiment has 2 LO's then,



Note: Color coding indicates that every LO should have task and assessment corresponding to it

So the crux of it is, every LO needs to have a task and assessment question. The task could be performing some action in the simulation and then attempting assessment questions in the form of MCQ's. For example, we could ask the student to set the values of a parameter through slider and observe the reading in an apparatus and then analyse the data to interpret the results. Using this we can figure out the engagement level of the student as well.

1.5 Simulator Interactions

Simulator interaction is the capability of the simulator to provide interactive features to the user. This lets the user explore the interactive capabilities of the simulator to enhance the learning outcome.

So simulator interactions serves two main aspects

1. Interactivity capabilities of the simulator
2. Enhancing learning outcome through interactivity and increasing engagement level of the user.

In the simulator interactions section a developer must provide information on:

What will the students do?

This section contains information on the inputs provided by the user or the student. This could be different inputs that would change from one simulation to another. For instance, it could be changing the value of a particular parameter, or it could be selecting some items from the drop down menu. The inputs provided by the user should be useful for getting output from the simulator and serve some meaningful purpose.

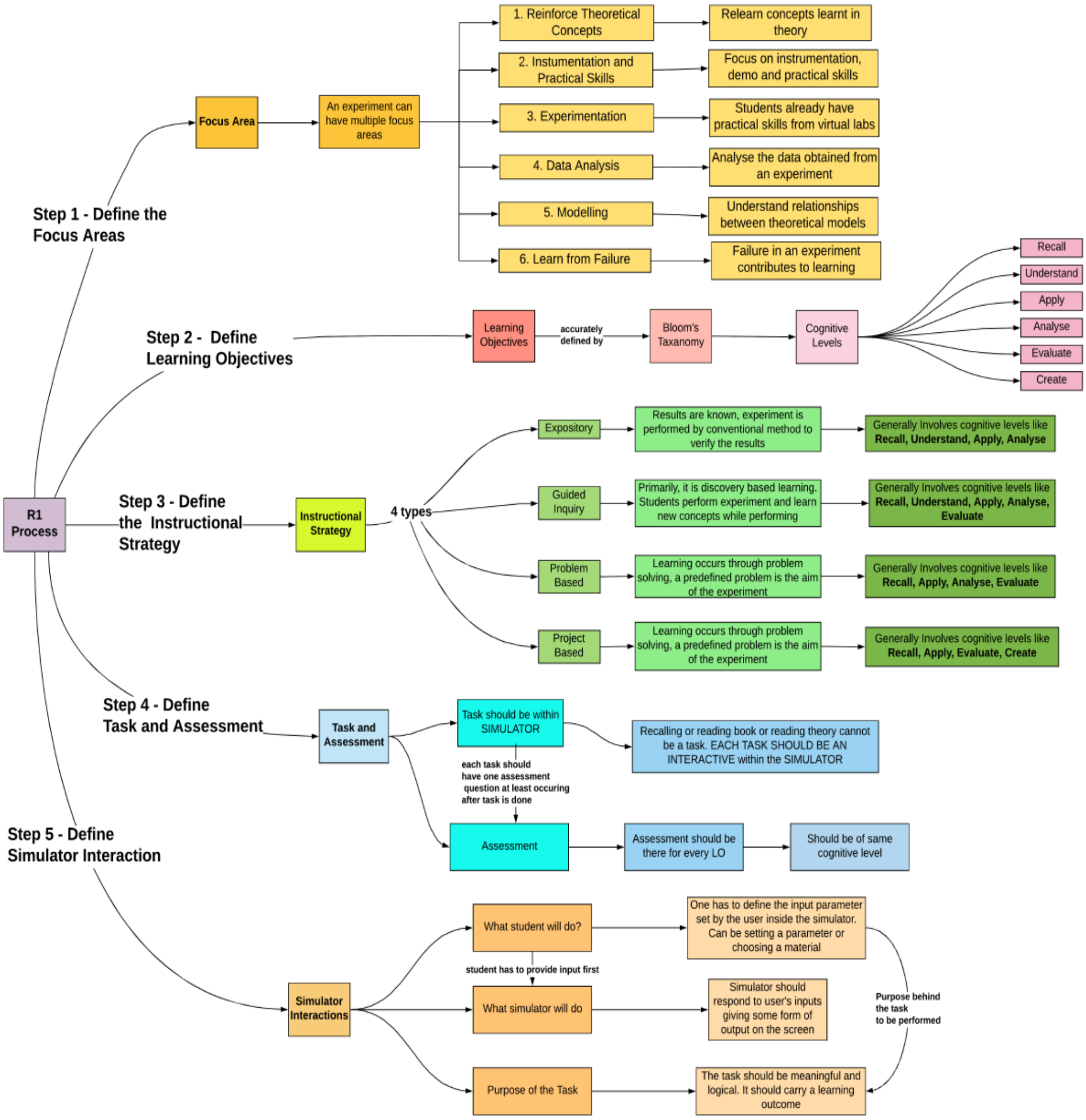
What will the simulator do?

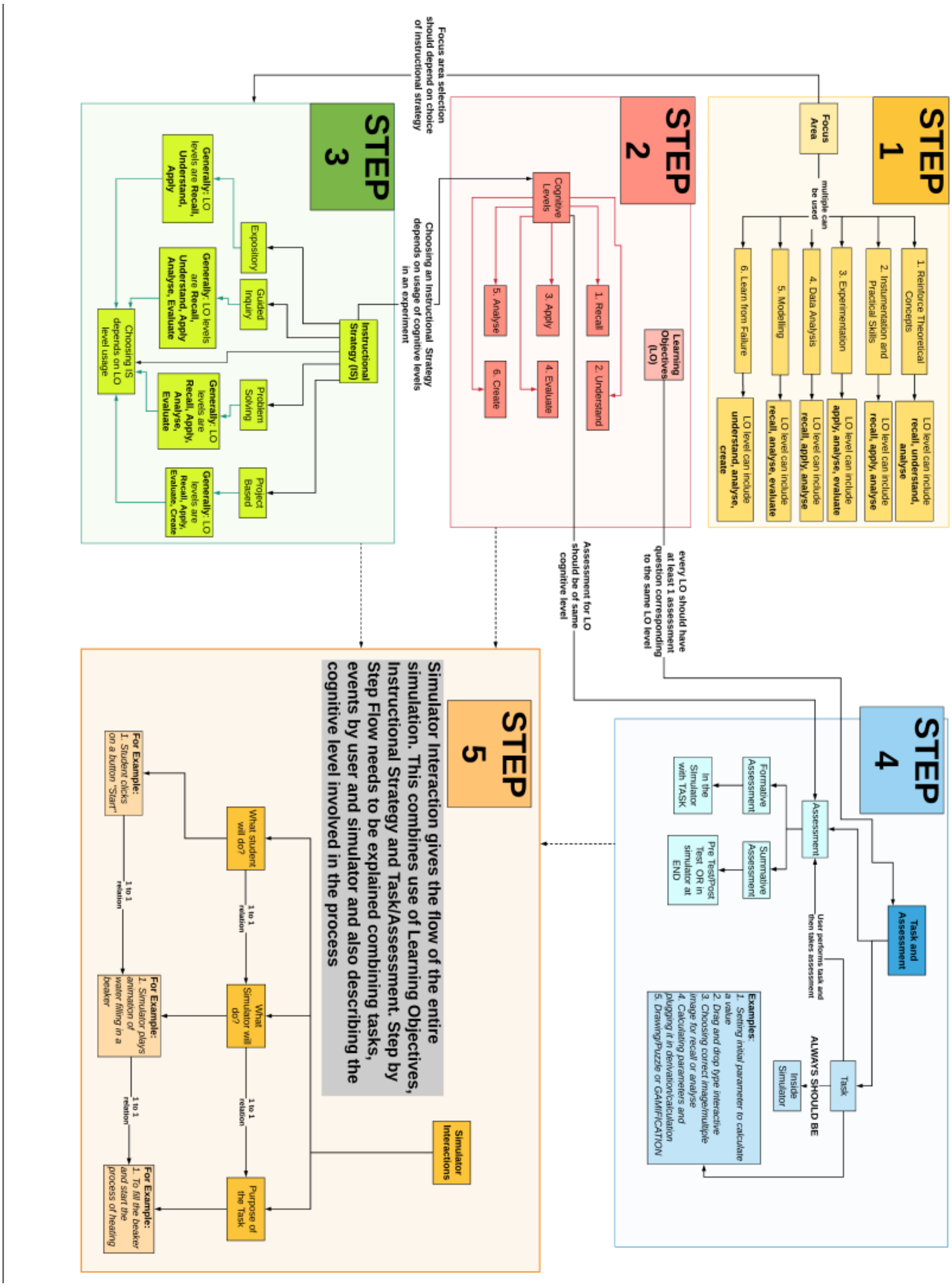
This section contains the information pertaining to the feedback that the simulator will provide depending on the inputs provided by the user. Note that the action by the user or the student should have one to one correspondence. One has to give the detailed information on the simulator's interactivity.

Purpose of the task?

The Main purpose of the task section one has to justify the purpose or the objective of the task the user does to get some interactive feedback from the simulator. This section mainly should explain how learning will happen through the student's input and simulator's feedback.

All the three sections have 1 to 1 relation and one must note that there should be logical and well reasoned philosophy behind the purpose giving each and every specifics.





Chapter 2: Strategies and Visualisation

2.1 Story Outline and Story:

Story Outline

The story outline should give a brief idea about the story. This could be 2 or 3 paragraphs.The story would contain all the information regarding the coverage of the virtual lab experiment.

Story

After writing the story outline, one has to now proceed to write an elaborative story. The story should be in simple English language. It should be logical and should make sense giving all the details of the coverage of the entire virtual lab experiment

The story should cover some key points, they are:

- ○ Set the Visual Stage description
- ○ Set the User objectives and goals
- ○ Set the pathway activities
- ○ Set Challenges and Questions/Complexity/ Variations
- ○ Allow Pitfalls
- ○ Conclusion
- ○ Equations/ Formulas

Set the Visual Stage description:

This is simply the stage on which the user will land on opening the simulation. One must mention all the visual elements seen by the user in the landing section. It could have different elements like buttons, sliders, drop down menus or some image with text.. One has to list them down in a simple language so that one can visualise how the landing page would look like

Set the User objectives and goals:

In this section, one should describe the objectives and goals of the experiment. The objectives and the goals should be written clearly and explain how the experiment would begin and what would be the flow of the experiment. This should not be confused with bloom's LO's as this is experiment specific objective and not concept specific LO.

Set the pathway activities:

Pathway means the path required to achieve a particular objective. So pathway activities implies how a user will traverse through the experiment. There could be several pathways on achieving the objectives but one should ensure that an effective pathway is chosen.

Set Challenges and Questions/Complexity/ Variations:

In this section one has to set challenges based on the user's level. By user's level it means, for set of weak students even a easy challenge can be difficult. So it is important to define the level of challenge properly. These challenges could be questions or problem solving. It would help the user in two ways, one is to invoke curiosity and interest and other to ensure learning and relearning.

Allow Pitfalls:

We often say that people learn from experiences. These experiences help clearing misconceptions, misunderstandings. The same thing holds true here, to "allow pitfalls" means to put the user in a situation where the user can make mistakes and then re-learn from it. This can be done in different ways, like allowing pitfalls in challenges or tricky questions. This would also help us understand the engagement factor of user.

Conclusion:

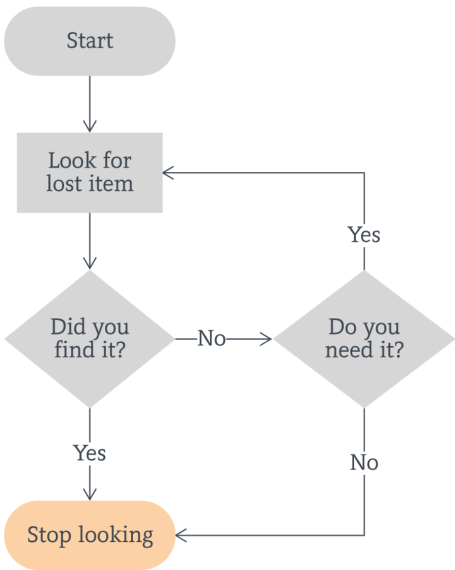
This may include displaying: how much time was taken?/ How many questions were right?/ How many hints did the learner use. Based on this the evaluation/marking to be suggested to the user. Once the user is towards the completion of the experiment, there should be descriptive interpretation based on the user's performance. For example, It could be the user has understood properly and which LO needs more effort

Equations/ Formulas:

These equations and formulas responsible for the working of experiment should be mentioned in a separate sheet. These equations could be written with a brief description about the dependent and independent variables and which equation is used where. Thus these formulas and equations basically govern the entire experiment and should be mentioned clearly so that one can easily troubleshoot in case of any difficulty

2.2 Flowchart

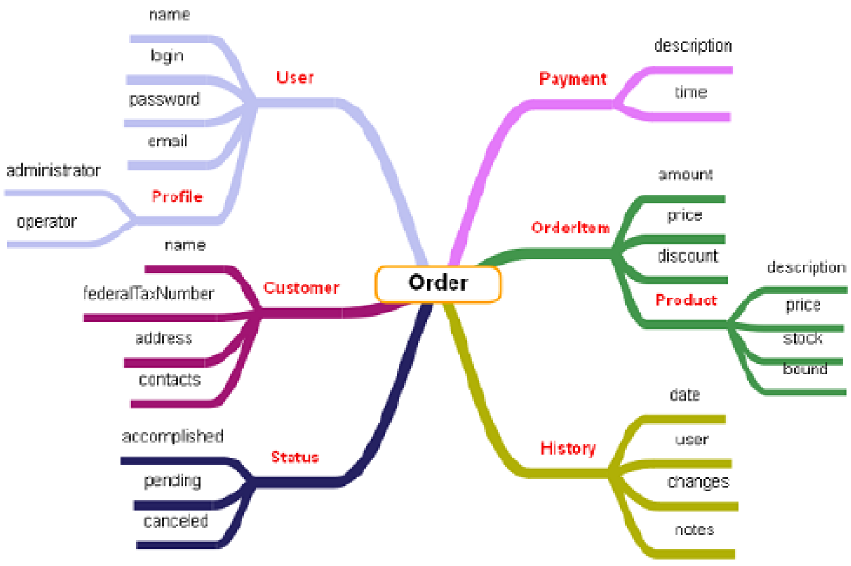
Flowchart is a diagram that helps us understand a process. For instance if you had a robot and want to set instructions for it. One simple way is to do it as shown in the figure.



The steps should be logical and reasoned. A flowchart basically gives us a logical flow and it is used in virtual lab experiments to define the process involved in the virtual lab experiment. The process could involve steps that has to be followed and instructions must be provided with a proper drawing. One can draw the flowchart and save the flowchart as .png.

2.3 Mindmap

A mind map is a diagram used to visually organize information. A simple mind map can look like this:

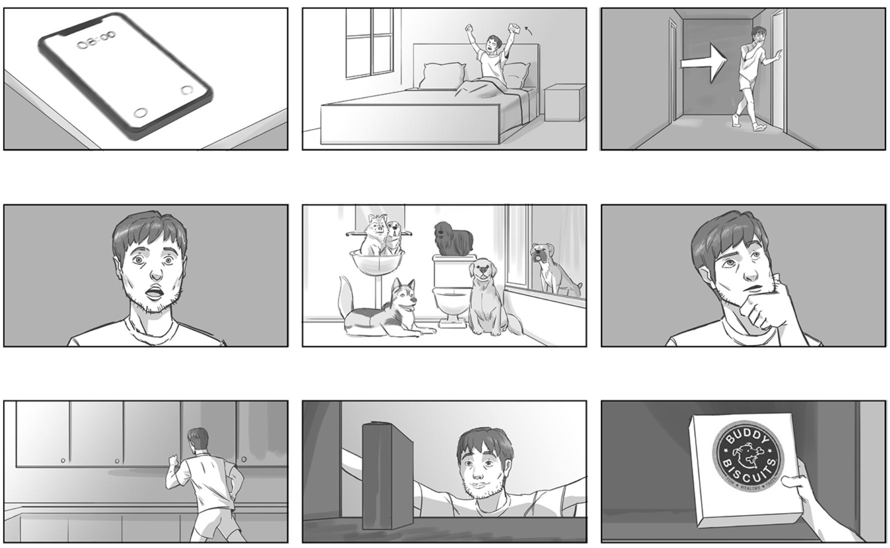


The main objective of mind map is to visually represent sub concepts linked to the main concept. The topic or the main topic is located at the center and is connected to subtopics independently. Furthermore these subtopics can be further connected to micro level of subtopics.

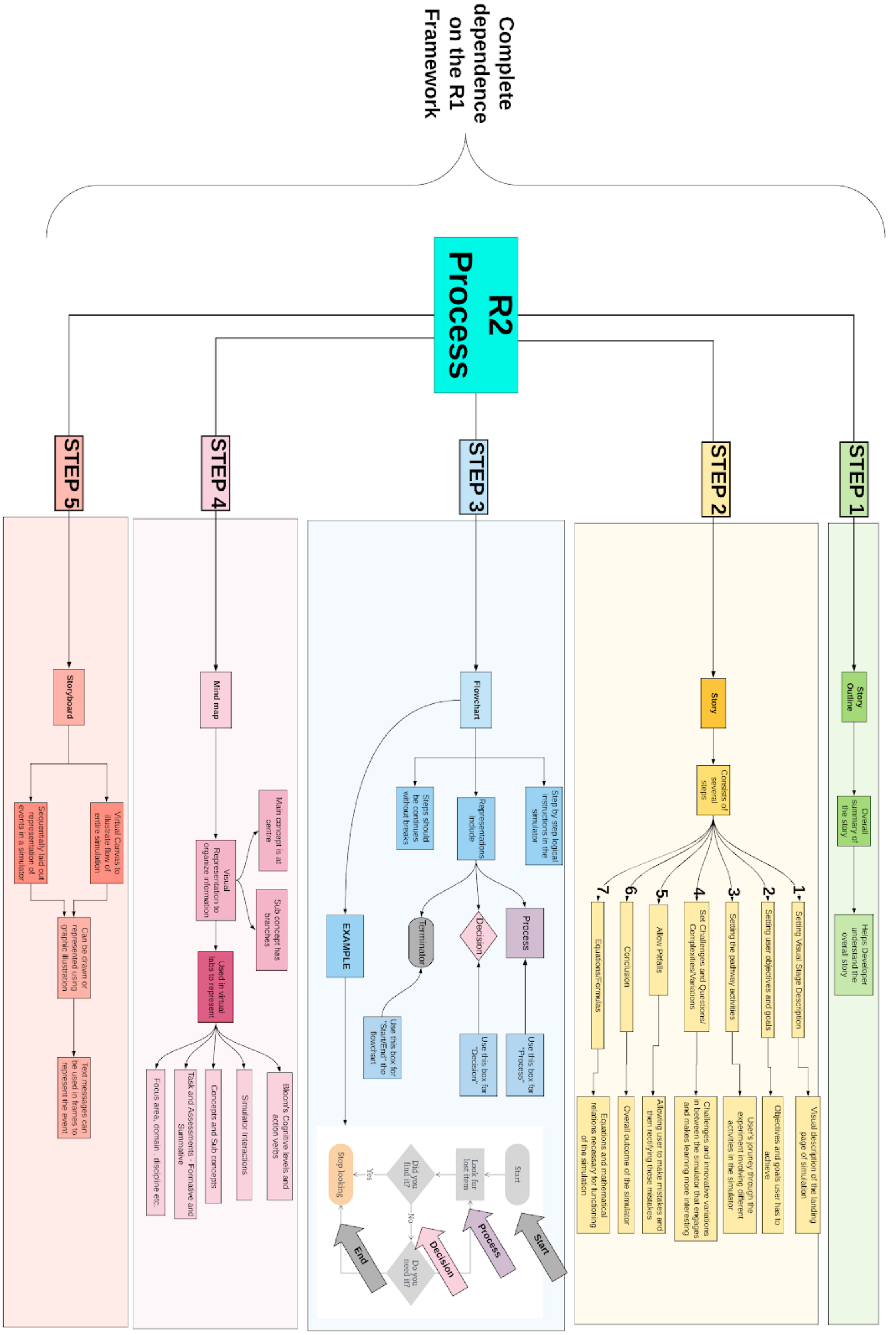
For example Heat ---measured by -----> thermometer ----- can be ----> gas thermometer.

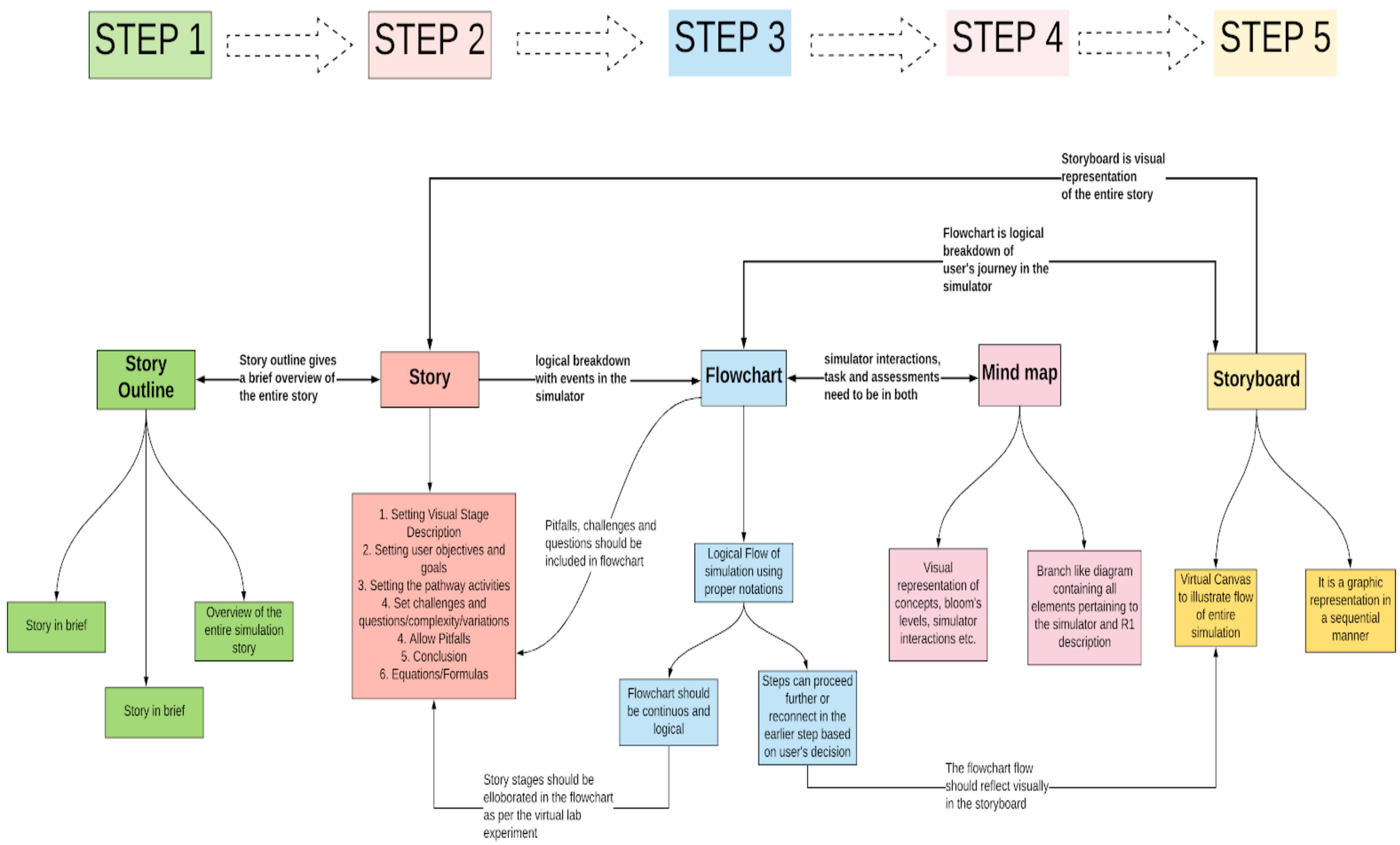
2.4 Storyboard

A storyboard is a graphic organizer in the form of illustrations or images displayed in sequence. It is used for the purpose of pre-visualizing a motion picture, or interactive media sequence.



In this case it is the virtual lab simulation. Making a general storyboard is very simple. Storyboard can be drawn frame by frame and what is to be present can be represented via cartoons, objects etc. One can make storyboard using https://wonderunit.com/storyboarder/. Or can use any other Open Source Image editing / drawing tool




STEP 1
STEP 2
STEP 3
STEP 4
STEP 5
Storyboard is visual representation of the entire story
Flowchart is logical breakdown of user's journey in the simulator
Story Outline
Story outline gives a brief overview of the entire story
Story
logical breakdown with events in the simulator
Flowchart
simulator interactions, task and assessments need to be in both
Mind map
Storyboard
Story in brief
Overview of the entire simulation story
Story in brief
1. Setting Visual Stage Description
2. Setting user objectives and goals
3. Setting the pathway activities
4. Set challenges and questions/complexity/variations
4. Allow Pitfalls
5. Conclusion
6. Equations/Formulas
Pitfalls, challenges and questions should be included in flowchart
Logical Flow of simulation using proper notations
Flowchart should be continuos and logical
Steps can proceed further or reconnect in the earlier step based on user's decision
Visual representation of concepts, bloom's levels, simulator interactions etc.
Branch like diagram containing all elements pertaining to the simulator and R1 description
Virtual Canvas to illustrate flow of entire simulation
It is a graphic representation in a sequential manner
Story stages should be elloborated in the flowchart as per the virtual lab experiment
The flowchart flow should reflect visually in the storyboard

Documentation on Git
Development Workshop / STTP

Git Documentation

3.1 Introduction

GitLab is a free and open source, web-based repository manager that lets teams collaborate on code, duplicate code to safely create and edit new projects, then merge finished code into existing projects.
It provides centralized, cloud-based locations where teams can store, share, publish, test, and collaborate on web development projects.
Its permissions, branch protection, and authentication features are what really make it stand out. Teams can secure projects on a more granular level, and projects are kept even safer while they're being worked on.

3.2 Advantages of GitLab

- A convenient user interface enables users to access everything from one screen: projects, latest projects, users, latest users, groups, and stats.
- Settings allow users to control whether a repository is public or private.
- "Snippet support" lets users share small pieces of code from a project, without sharing the whole project.
- Protected branches are a new way to keep code safe. They allow users to set higher permissions on a project, so only certain people are able to push, force push, or delete code in a branch.
- Authentication levels take this security a step further, allowing users to give people access beyond a read/write level. For example, you can give a team member access to issue tracking without having to give them access to the code itself.
- Improved milestones enable you to set milestones at a group level, not just a developer-specific level. Developers can get insight into the whole team's scope and view the entire project's milestones, not just their own.
- With the "Work in Progress" status, developers can label a project "WIP" to let collaborators know that the code is unfinished. This prevents it from accidentally getting merged with other code before it's finished.
- You can attach files like comments to any communications in GitLab.

3.3. Installing Git

3.3.1 On Windows:

1. Download the latest Git For Windows Installer.

2. When you've successfully started the installer, you should see the Git Setup wizard screen. Follow the Next and Finish prompts to complete the installation. The default options are pretty sensible for most users.

3. Open a Command Prompt (or Git Bash if during installation you elected not to use Git from the Windows Command Prompt).

4. Run the following commands to configure your git username and email using the following commands. These details will be associated with any commits that you create:

```
git config -–global user.name “Your Name”
git config -–global user.email “Your Email”
```

3.3.2 On Linux:

1. From your shell, install Git using apt-get:

```
sudo apt-get update
sudo apt-get install git
```

2. Verify the installation was successful by typing

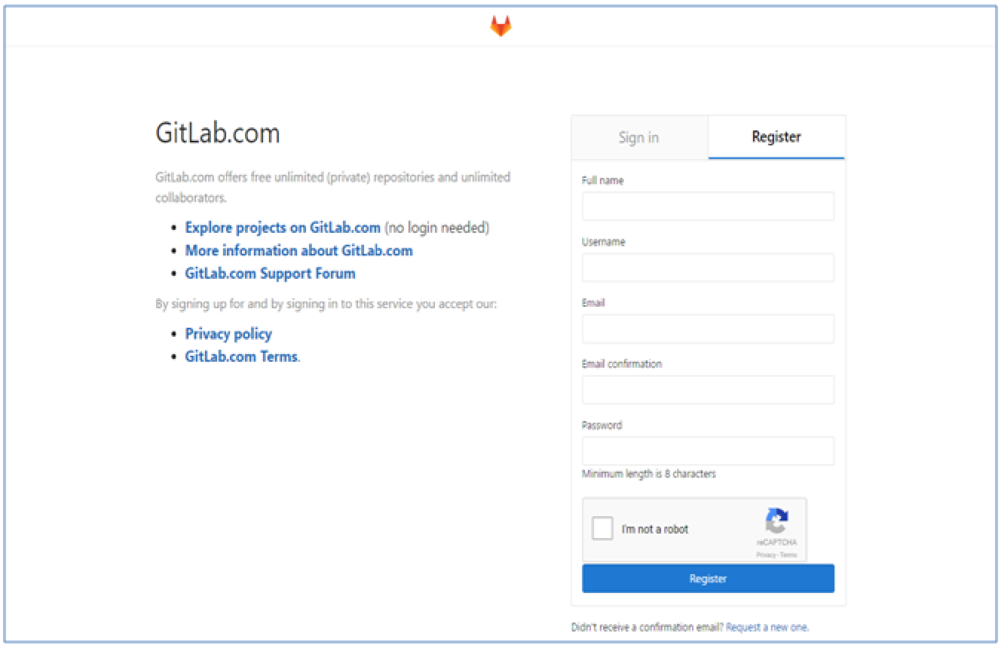
```
git --version
```

3. Configure your git username and email using the following commands. These details will be associated with any commits that you create:

```
git config –global user.name “Your Name”
git config –global user.email “Your Email”
```

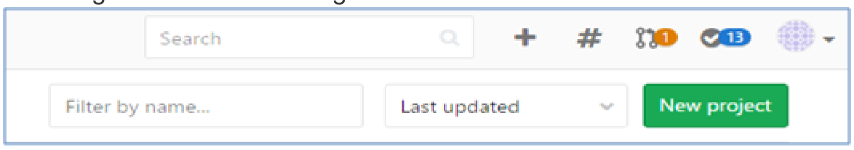
3.4. GitLab Account Creation

1. To create a gitlab account, go to https://gitlab.com/users/sign_in
2. Click on Register, fill the details and register.

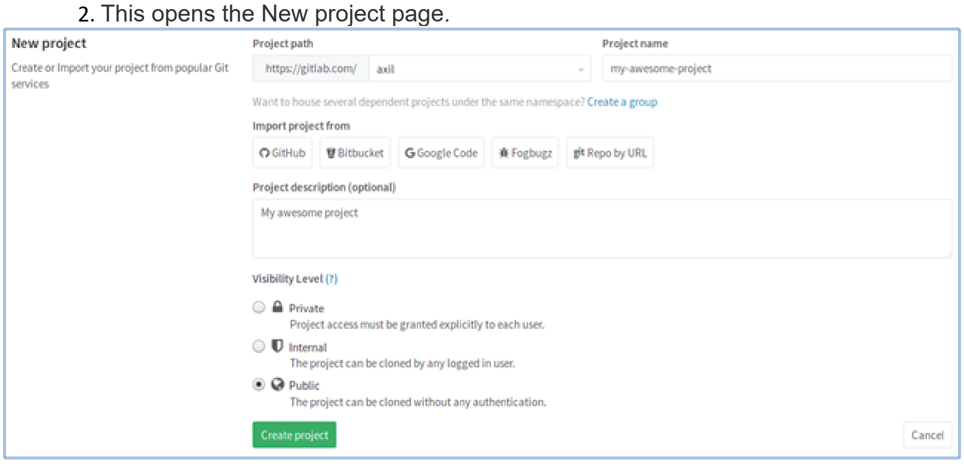


3.5. Create project in GitLab

1. In your dashboard, click the green New project button or use the plus icon in the upper right corner of the navigation bar.



2. This opens the New project page.



3. Provide the following information:

a. Enter the name of your project in the Project name field. You can't use special characters, but you can use spaces or hyphens.

b. If you have a project in a different repository, you can import it by clicking an Import project from button provided this is enabled in your GitLab instance.

c. The Project description (optional) field enables you to enter a description for your project's dashboard, which will help others understand what your project is about.

d. Changing the Visibility Level modifies the project's viewing and access rights for users.

4. Click Create project.

3.6. Git init:

Git init will initialize new repository.

Command Line Instructions

- Go to your computer's shell and type the following command:

git init

An empty repository will be created in the working directory.

3.7. Cloning the project

In Git, when you copy a project you say you "clone" it. To work on a git project locally (from your own computer), you will need to clone it. To do this, sign in to GitLab.

When you are on your Dashboard, click on the project that you'd like to clone. To work in the project, you can copy a link to the Git repository through a SSH or a HTTPS protocol. While you are at the Project tab, select HTTPS and copy the link using the 'Copy to clipboard' button (you'll have to paste it on your shell in the next step).

Command Line Instructions

- Go to your computer's shell and type the following command:

git clone PASTE HTTPS HERE

A clone of the project will be created in your computer.

3.8. Adding changes to the project

To add the changes in the project, git add is used.

Command Line Instructions

- Go to your computer's shell and type the following command:

git add .

The changed files will be added to the project.

3.9. Committing changes to the project

To commit the changes in the project, git commit is used.

Command Line Instructions

- Go to your computer's shell and type the following command:

git commit –m "Some message"

The changed files will be committed to the project.

3.10. Git status:

Shows the working tree status.

Command Line Instructions

- Go to your computer's shell and type the following command:

git status

Status of working tree will be displayed.

3.11. Git fetch:

Git fetch command will download objects and refs from another repository.

- Go to your computer's shell and type the following command:

git fetch

Branches and/or tags (collectively, "refs") from one or more other repositories, along with the objects necessary to complete their histories will be fetched.

3.12. Git rebase:

git-rebase command will reapply commits on top of another base tip.

- Go to your computer's shell and type the following command:

```
git rebase <branchname>
```

This will show whether the branch is up-to-date or not.

3.13. Git pull:

Git pull command will Fetche from remote repository and integrates with another repository or a local branch.

Command Line Instructions

- Go to your computer's shell and type the following command:

```
git pull origin <repository name>
```

Changes from a remote repository will be incorporated into the working branch.

3.14. Pushing project to main repository

Uploading of project to main repository is done with git push.

Command Line Instructions

- Go to your computer's shell and type the following command:

```
git push origin master
```

- Enter the password if asked.
- The changed files will be pushed to the original project.

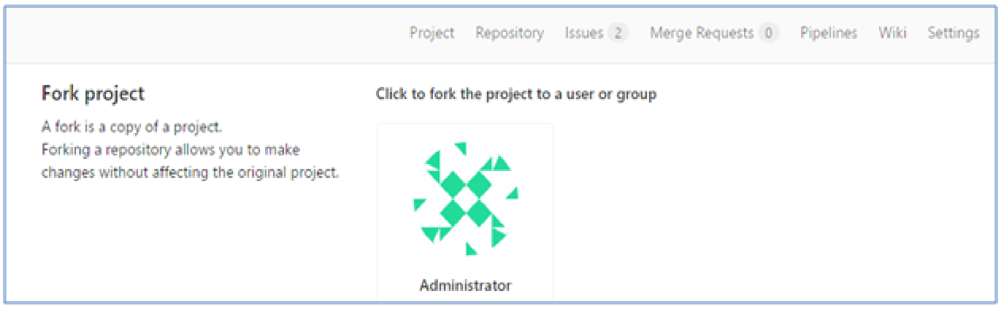
3.15. Forking a project

A fork is a copy of an original repository that you can put in another namespace where you can experiment and apply changes that you can later decide if publishing or not, without affecting your original project.

It takes just a few steps to fork a project in GitLab.

1. Go to a project's dashboard under the Project tab and click on the Fork button.

2. You will be asked where to fork the repository. Click on the user or group to where you'd like to add the forked project.



3. After a few moments, depending on the repository's size, the forking will complete.

3.16. Branching a project

Branching offers a way to work on a new feature without affecting the main codebase. Branches serve as an abstraction for the edit/stage/commit process discussed in Git Basics.

Each time, that you want to commit a bug or a feature, you need to create a branch for it, which will be a copy of your master branch.

When you do a pull request on a branch, you can continue to work on another branch and make another pull request on this other branch.

Command Line Instructions

- Go to your computer's shell and type the following command:

git checkout -b <branch_name>

The branch of the master project will be created.

3.17. MISCELLANEOUS

If you want to check your settings, you can use the git config --list command to list all the settings Git can find at that point:

$ git config --list

user.name=John Doe
user.email=johndoe@example.com
color.status=auto
color.branch=auto
color.interactive=auto
color.diff=auto
...

You may see keys more than once, because Git reads the same key from different files (/etc/gitconfig and ~/.gitconfig, for example). In this case, Git uses the last value for each unique key it sees.

You can also check what Git thinks a specific key's value is by typing git config <key>:

$ git config user.name
John Doe

For changing Credential

git config --global credential.helper osxkeychain
You'll then be prompted for your password again.
For Windows, it's the same command with a different argument:

git config --global credential.helper wincred

3.18. IN a Nutshell

git config —global user.name "Your Name"
git config —global user.email "Your Email"

1. $ git clone URL
2. git add .
3. $ git commit -m "commit Message"
4. $ git status (optional step)
5. $ git push origin master
6. $ git pull

3.19. Git Structure:

1.Working Directory:
This is the place where we can work with the files and folders which are all pulled out from the remote repository.
2.Virtual directories:
Below are the virtual directories which are acting as the supporting environment for maintaining the revision with the git repository.
3.Index (Cache) or Staging:
This is handling the set of file changes being collected (with "git add") to create in the repository as a single commit. "Staged Changes" are referring to any changes made on file and added to the index (with "git add") to create in the repository as a single commit. "Unstaged Changes" are referring to the file changes made but not (or not yet) added to the index.
4.Local Repository:

This component is playing as the virtual local repository which will be acting as the mirror of our remote repository. This component is more useful for pushing our changes, taking the difference between the working directory and local repository etc.
5.Remote Repository:
This is the server where all the files and directories are resides and those can be tracked from different networks and allowing multiple users to take role on the same files and directories.

Appendix

Guidelines to design effective virtual labs – Implementation Level

Guidelines to design an Expository laboratory experiment

For decades, the dominant laboratory instructional style has been expository laboratories in which the instructor defines the topic to be investigated, provides a context for the investigation, has students repeat instructions or follow them from a manual, and then compare results with a predetermined outcome known to both student and instructor. This instructional style gives little emphasis to critical thinking and conceptual change, which is unfortunate given the current consensus that conceptual learning is the most important outcome of laboratory instruction. It has been suggested that laboratory instructional style may affect learning, with a particular laboratory format being better suited for meeting a given goal. For example, conceptual learning may benefit from an inquiry-based or questioning approach, whereas skills are best taught by more direct instructional techniques.

The main components of the expository laboratory instruction are as follows:

Aim/Purpose: What is the reason for doing the experiment?

Learning Objectives: What is there to be learned from doing the experiment?

Hypothesis: What "you think" will be the final outcome of the experiment. This is generally based on prior knowledge or observations. In other words, you are not just pulling this "out of thin air"; you have some logical reason for thinking this. If you have no prior knowledge of the concept, you will need to do research before making a hypothesis. Also, explain exactly "why you think this".

Apparatus/Materials: A list of equipment and supplies that will be needed to complete the lab procedure. List the major pieces of equipment first.

Procedure

1. The step by step process that is followed in carrying out the experiment. Preferably, the steps are sequentially listed in the order they need to be followed to complete the experiment successfully. Be very exact with each step in case someone else wants to repeat your procedure. It's like telling someone how to find your house. The least little mistake or detail left out could be critical to the outcome of the experiment.
2. Using the safety symbols, identify any precautions that may need to be followed in completing this experiment.
3. Identify the variables in the experiment:
    - Controlled - factors that remain constant throughout the experiment.
    - Independent - the one factor that will be manipulated or changed during the experiment.
    - Dependent - the variable that becomes altered as a result of the change that was made in the independent variable.

Observations:

Observations are of two categories:

- Qualitative - information gathered through the senses such as smell, taste, touch, hear, shape, etc.

- Quantitative - information gathered due to precise measurements, such as height in cm, width in cm, mass in g, volume in cm3, density in g/cm3, time in seconds, speed in kph, etc.

Observations are organized in:

- Data tables or charts.

Results and Conclusion

Graphs are visual representations of the data so that it can be easily studied, interpreted, and analyzed. Circle, bar, and line are examples of kinds of graphs. This is a written summary of what was actually learned from doing the experiment. The conclusion will either support or reject the proposed hypothesis.

Examples of Expository Laboratory Experiment Design

Sample Laboratory Experiment Design #1

Construct the given circuit. Measure the current flowing through the diode at various values of applied DC voltage. Note down the readings for ten values. Plot the graph of current Vs. voltage to obtain the V- I Characteristics of the PN junction diode. Calculate the static and dynamic resistance of the diode from the formulae given.

Guidelines to design a Discovery or Guided Inquiry based laboratory experiment

Inquiry means that students are handling science; they are manipulating it, working it into new shapes and formats, integrating it into every corner of their world, and playing with it in unknown ways. Inquiry implies that students are in control of an important part of their own learning where they can manipulate ideas to increase understanding.

"Scientific inquiry refers to the diverse ways in which scientists study the natural world and propose explanations based on the evidence derived from their work. Inquiry also refers to the activities of students in which they develop knowledge and understanding of scientific ideas, as well as an understanding of how scientists study the natural world."

Alan Colburn suggests three levels in inquiry based learning – Structured Inquiry, Guided Inquiry and Open Inquiry. Discovery learning or Guided Inquiry is the most fundamental form of inquiry-oriented learning. The focus of discovery learning is not on finding applications for knowledge but, rather, on constructing concepts and knowledge from experiences. As such, discovery learning employs reflection as the key to understanding. The teacher introduces an experience in such a way as to enhance its relevance or meaning, uses a sequence of questions during or after the experience to guide students to a specific conclusion, and questions students to direct discussion that focuses on a problem or apparent contradiction. Employing inductive reasoning, students construct simple relationships or principles from their guided observations.

Phases of Inquiry

According to Nancy T. Davis, Associate Professor of Science Education at Florida State University (personal interview, 2002), there are four phases within each of the inquiry approaches. Each phase is characterized by questions that guide students to make their own discoveries.

Initiation Phase

The Initiation Phase is the first phase in all levels of inquiry. It is primarily designed to stimulate and motivate students' curiosity through questioning. This phase provides students with an opportunity to experience a phenomenon or something new that challenges a previous belief or assumption.

- Have you ever seen…?

- Did you notice…?

- What did you observe…?

Exploration Phase

The Exploration Phase is the second phase of inquiry. In this phase, questions are eliminated or narrowed down to those types of questions students can actually physically answer through experimentation or research.

- What happened when…?

- What did you…?
- What could we do to find out…?
- What questions do you have…?

Experimentation Phase

The third phase of inquiry is the Experimentation Phase. This is where students form into groups to conduct an experiment. Students collect data and information, and then formulate a method of presentation.

- What did you find out about…?
- How is it the same as or different from…?
- What do you know about the characteristics of…?

Presentation Phase

The last phase of inquiry is the Presentation Phase. Groups or individuals take the information gathered in the experiment and put it into some form of presentation. *PowerPoint*presentations or project display boards are types of presentations that may be used. The group or individual will share the data with an audience and allow time for questions concerning procedures, data, information, etc.

- Can you explain why…?
- Why do you think…?
- What other factors may be included in…?
- Can you find a way to…?
- How did you arrive at a solution to…?

Problems in Inquiry Instructions:

- Provide students enough time to complete an investigation.
- Be aware that often students have difficulty identifying a problem to investigate.
- Students may be disinterested, apathetic, or will not engage in the inquiry.
- Be aware that students may have difficulty mastering or designing procedures.

Examples of Guided Inquiry/Discovery Laboratory Experiment Design

Sample Laboratory Experiment Design #1

The experiment design should have the following components:

- Focus Question:
- Learning Objectives:
- Background Information
- Experimental Procedure
- Materials needed
- Guided Inquiry list of questions

Example:

Focus Question: How does the resistance of a PN Junction diode vary?

Procedure: To carry out this analysis what procedure will you follow? What circuit will you use? What are the observations necessary to carry out the experiment? What data will you gather? How will you represent the data? What type of data analysis will you carry out? What can you infer from the analysis of the data? Do the results answer the focus question? What can you conclude from the results?

Sample Laboratory Experiment Design #2

THE JUNKYARD

In a junkyard, Mr. X, the owner of the junkyard, uses big electromagnet cranes to move old cars. The electromagnet crane he uses is able to lift standard-sized cars easily. However, Mr. X has realized that it is not appropriate for holding bigger cars, such as SUVs, which are two times heavier than usual cars. Because Mr. X cannot afford to buy a new electromagnet crane, he tries to make a more powerful electromagnet crane by doing some changes using the materials that are available in his junkyard.

- Can you give some advice to Mr. X on this issue as a physicist?
- What can be the physics concepts related to this situation?
- What is the main problem of Mr. X when you look at from a physicist's perspective?
- Write a hypothesis that may provide a solution to the problem:
- Design an experiment to test your hypothesis in the laboratory:
- What are the variables that can affect your experiment?

Manipulated:, Responding:, Controlled:, Uncontrolled:

Your results and observations:

Interpretation of your results and observations: What is your recommendation to Mr. X?

Guidelines to design a structured problem solving laboratory experiment

Well-structured problems are problems that always use the same step-by- step solution. The characteristics of well-structured problems are: Solution strategy is usually predictable, convergent (one right answer) and all starting information is usually part of the problem statement. So the experiment design for such a problem solving is simple. The structured problem is given to the students as the laboratory exercise and the students are required to come up with the solution using the experimental procedures, collect data, analyse the data and arrive at the desired solution.

Examples of Structured problem solving Laboratory Experiment Design

Sample Laboratory Experiment Design #1

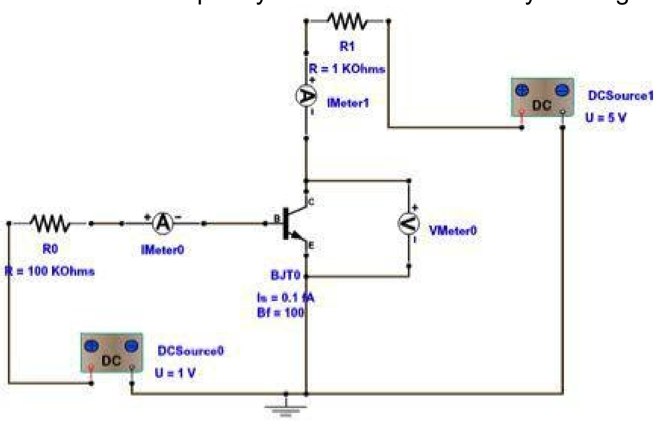
In the CE Transistor circuit shown VBB= 5V, R0= 107.5 k Ω, R1 = 10 k Ω,

VCC = 10V. Construct the circuit and obtain the DC Characteristics.

Find IB, IC, VCE, β and the transistor power dissipation using the characteristics.

- Compare the theoretical and practical values obtained.

- Also compare your values with that of your neighbour.



Sample Laboratory Experiment Design #2

Design a Mod 8 counter, implement the design in the virtual lab and obtain the output waveforms.

Guidelines to design a Problem-based laboratory experiment

Problem-based learning(PBL) is a student-centered approach in which students learn about a subject by working in groups to solve an open-ended problem. The problem is what drives the motivation and the learning.

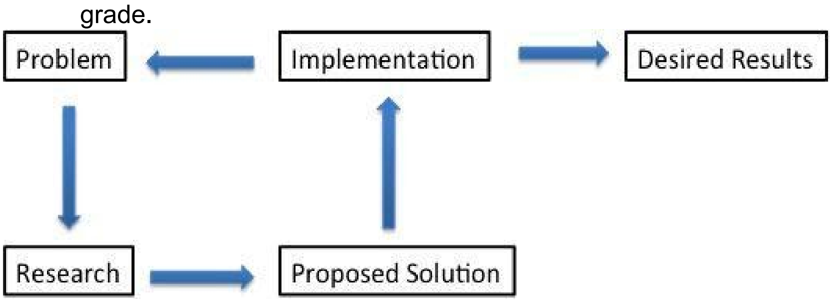
Rather than teaching relevant material and subsequently having students apply the knowledge to solve problems, the problem is presented first. Students generally must:

1. Examine and define the problem.
2. Explore what they already know about underlying issues related to it.
3. Determine what they need to learn and where they can acquire the information and tools necessary to solve the problem.
4. Evaluate possible ways to solve the problem.
5. Solve the problem.
6. Report on their findings.

PBL assignments can be short, or they can be more involved and take a whole semester.

The basic steps in designing PBL experiment/project are:

- Articulate the learning outcomes of the project. What do you want students to know or be able to do as a result of participating in the assignment?
- Create the problem. Ideally, this will be a real-world situation that resembles something students may encounter in their future careers or lives. Cases are often the basis of PBL activities.
- Establish ground rules at the beginning to prepare students to work effectively in groups.
- Establish how you will evaluate and assess the assignment. Consider making the assessments students make of their own work and that of their peers part of the assignment grade.



The figure illustrates the steps in a problem based laboratory experiment.

Examples of Problem-based Laboratory Experiment Design

Sample Laboratory Experiment Design #1

1. An assembly line has 3 failsafe sensors and 1 emergency shutdown switch. The Line should keep moving unless any of the following conditions arise:
   - If the emergency switch is pressed.
   - If sensor 1 and sensor 2 are activated at the same time.

- If sensor 2 and sensor 3 are activated at the same time.

- If all three sensors are activated at the same time.

(a) Derive the truth table for this system.

(b) Design, using Karnaugh Map techniques, a minimum AND-OR gate network for this system.

(c) Draw the resulting digital circuit diagram.

(d) If the time delay experienced by a NAND gate is 8ns and the time delay experienced in a NOR gate is 10ns. Which implementation of (c) is faster? By how long?
(c) Design, a digital circuit that will implement the minimal AND-OR gate network found in (b) using both

(i) NAND gates only

(ai) NOR gates only.

Assume that each logic gate can have any number of inputs and that inverted inputs are available.

Sample Laboratory Experiment Design #2

A warning buzzer is to sound when the following conditions apply:

- Switches A, B, C are on.
- Switches A and B are on but switch C is off.

- Switches A and C are on but switch B is off.

- Switches C and B are on but switch A is off.

1. Draw a truth table for this situation and obtain a Boolean expression for it.

2. Minimize this expression and draw a logic diagram using only a) NAND b) NOR gates.

3. If the delay of a NAND gate is 15ns and that of a NOR gate is 12ns, which implementation is faster.

Sample Laboratory Experiment Design #3

A bank has 3 locks with a key for each lock. Each key is owned by a different person. In order to open the vault door at least two people must insert their keys into the assigned locks at the same time. The trainee (i.e. person 3) can only open the vault when the manager (i.e. person 1) is present in the opening. The signal lines (A,B,C) are 1 if the key is inserted into locks 1,2,3 respectively.

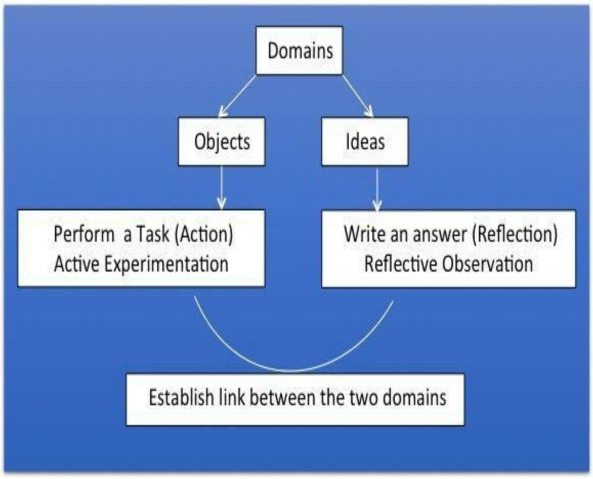
Determine the truth table for such a digital locking system

Design, using Karnaugh Map techniques, a minimum AND-OR gate network to realize this locking system. Implement the minimal circuit found in the previous section using: two input NAND gates only / Two input NOR gates only.

If a two input NAND has a delay of 1.5ns, a two input NOR gate has a delay of 1.1ns, which of the locking system implementations is the fastest.

Cognitive structure of the tasks

Labwork includes a wide variety of tasks, designed to promote quite different kinds of learning. The fundamental purpose of any labwork task is to help students to make links between two domains: the domain of real objects and observable things, and the domain of ideas. When the labwork task is implemented we can observe what the students actually do on the task, and we can attempt to assess what they actually learn.



Tasks to be performed by the students based on Teacher's instructions. The instructions and the tasks should be matched to the Learning Objectives.

Sample for Round 1

Title: Experiment Design for Fluid Mechanics Lab –

Discipline	Chemical, Mechanical, Civil	
Lab	Fluid Mechanics	
Experiment	Friction in cirular pipe	
Focus area	Experimental Analysis Skill	

Learning Objectives and Cognitive Level

Sr. No	Learning Objective	Cognitive Level
1. Students will be able to:	Identify different pipe sections and meaning of Reynolds number and friction factor	Recall
2. Students will be able to:	Identify instruments used in experiment such as test section, manometer, rotameter,	Recall

	variable speed centrifugal pump, water reservoir etc	
3. Students will be able to:	Describe how water flows from water reservoir to circular pipe through pump by turning on the switch, moving valve.	Understand
4. Students will be able to:	Explain how the flow rate changes with the movement of regulator	Understand
5. Students will be able to:	Explain the functionality of manometer by connecting it with pressure tapping lines.	Understand
6. Students will be able to:	Describe functionality of the rotameter	Understand
7. Students will be able to:	Describe the theoretical relation between Reynolds number and friction factor	Understand
8. Students will be able to:	Calculate average viscosity of water by using rotameter reading	Apply
9. Students will be able to:	Apply readings of manometer in the calculation pressure drop	Apply
10. Students will be able to:	Calculate Reynolds number and friction factor using observed readings	Apply
11. Students will be able to:	Apply calculated readings to plot the graph of log of Reynolds number v/s log of frictional factor.	Apply
12. Students will be able to:	Examine the difference between observed frictional factor and theoretical predicted frictional factor	Analyze
13. Students will be able to:	Analyse relation between Reynolds number and frictional factor by observing graph	Analyze
14. Students will be able to:	Assess the parameters which are responsible to establish relation between Reynolds number and frictional factor	Evaluate

Instructional Strategy Method: Expository

Assessment Method: Formative Assessment

Task & Assessment Questions:

Instructions given by the Teacher	Tasks to be done by the Students
Display/show the experimental set-up. Explain the significance of this experiment in real world applications. Mention the dimensions of the test section – diameter, length, distance between the pressure tappings and the properties of the fluid.	Retrive/Recall the various equipment and instruments involved in the experiment. Understand the significance of the diameter, length and pressure drop on the flow.
Highlight the set/initial values or status of the experiment. The knob to change the RPM should be in closed position and the valve should be in closed position. The level of the manometric fluid in both the limbs should be the same.	Ensure that these conditions are addressed or satisfied before starting the experiment.
Explain the procedure by first opening the valve and slowly changing the knob to increase the RPM.	Follow the instructions. Set the position of the valve to fully open position. Change the regulator knob slowly to the maximum position wherein the speed of the pump will be maximum.
The fluid used in the manometer and its physical properties and the pressure tappings. Ensure that the speed should be increased in such a way that the manometric fluid should not leave (escape) from the limb.	The position of the knob should be controlled so as to address the requirement and the height difference should be noted down.
Observations to be noted down.	Note down the flow rate displayed by the rotameter and the height difference based on the manometer.
Repeat the experiment for at least 10 – 12 different flow rates.	Change the position of the regulator knob in descending order to have at least 10 – 12 data points. Note down the height difference in the manometer for each flow rate.
Calculation of cross sectional area, velocity, pressure drop, Reynolds' number and friction	Calculate the flow area using the given correlation. Calculate velocity based on volumetric flow rate and cross sectional area.

factor based on the observations. Calculate friction factor based on theoretical correlations.	Calculate the pressure drop based on the height difference and properties of manometric fluid. Calculate the Reynolds' number. Calculate the friction factor using given equation. Calculate theoretical friction factor based on the given correlations depending on the Reynolds' number.
Validation	Make necessary changes/corrections in the calculations if the same are incorrect.
Comparison of friction factors	Compare experimental and theoretical friction factors.
Plot of friction factor vs Reynolds' number	Draw plot of log of observed and theoretical friction factors on Y-axis and log of Reynolds' number on X-axis.
Nature of graph	Comment on the nature of plot citing suitable reasons.

Additional Assessment Questions

The rotameter is mounted in _______ position.	Horizontal Vertical
The flow rate can be finely controlled by manipulating _______ valve of the pump.	Discharge Delivery
The manometer used is of ________ type.	Inclined tube U-tube Horizontal
Friction factor arises due to	Roughness of pipe Thickness of pipe Length of pipe
The values of observed friction factor will be ______ than theoretical friction factor.	Greater Less

Simulator – User interface and the Purpose

This includes the action of the simulator and the task to be performed by the students and the purpose of this action.

What Simulator will do ?	What students will do?	Purpose of the task
Display label image and display "Show initial condition" button	See the labeling of the set-up and click on the button.	To recall the set-up
Display initial conditions and display "Start Simulation" button	Understand initial condition and click on the button.	To provide basic environment to start the experiment.
Display basic experimental arrangement with the closed valve	See the basic arrangement and start the experiment by opening the valve.	Initiate the interaction with the experimental setup.
Display the Slider with the initial current position	Move the handle of the slider and observe the appropriate changes in Manometer and Rotameter	Change the speed of flow of water and generate the different situations
Display the "Add to Readings" Button	Click on the button	To save the current readings
Show the "Display Table" button	Click on the button	To observe the saved readings.
Display the "Reset" Button	Click on the button	To reset the situation with the initial conditions
Display the "Result" tab button	Click on that tab	To calculate the data based on readings
Display the empty table	Fill that table according the saved readings	To make the students calculate the data
Display the "Submit" button	Click on the button	To check for the calculated data whether it is correct or not
Display the "Generate Graph" button	Click on the button	To generate the graph according to the readings
Display "Generate PDF" button	Click on the button	To generate PDF which contains all the calculated and observed data

Practical context of task -- Duration of task

The duration of the task depends on the nature of the tasks. The average duration of the tasks assigned to the students in the electronics engineering virtual laboratories in order to achieve the learning outcomes selected in our context is arrived after the evaluation of the lab work. This is discussed in the various studies.

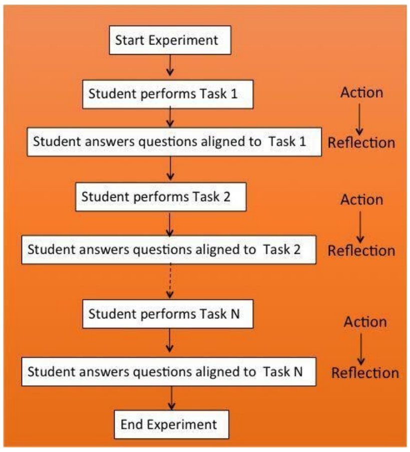
Task profile	Learning outcome criteria						Level of technical skills			Instructional style				Cognitive structure				
	R	I	E	DA	M	LFF	L	M	H	E	I	D	P	A	B	C	D	E
Duration of task	15	20	25	20	20	25	15	20	25	15		20	25	25	20	15	15	10

Guidelines for the Task Design based on Instructional strategy

Components of Experiment	Expository Instruction	Inquiry Instruction	Discovery Instruction	Problem-based instruction
Broad goal/objectives	Given	Given	Given	Given
Background Theory	Given	Given	Given	Given
Procedure	Given	Not given but decided by students	Not given but decided by students	Not given but decided by students
Circuit/System Design	Given	Not given but decided by students	Not given but decided by students	Not given but decided by students
Information about what data to be collected	Given	Not given but decided by students	Not given but decided by students	Not given but decided by students
Data Analysis and Results	Given	Not given but decided by students	Not given but decided by students	Not given but decided by students
Interpretation of Results	Not given but mostly known to the students	Not given but decided by students	Not given but decided by students	Not given but decided by students
Conclusion	Not given but mostly known to the students	Not given but decided by students	Not given but decided by students	Not given but decided by students

Guidelines for designing assessment questions aligned to learning objectives and tasks

After a comprehensive literature analysis we recommend that the type of assessment method, which should be followed in the virtual laboratories, is the Formative assessment or assessment for learning. This methodology can be followed as shown in the figure.



The assessment questions you ask should be aligned to the learning objectives and the tasks assigned to the students. The following table gives the list of assessment questions you can ask aligned to the tasks and learning objectives.

Tasks	Assessment Questions
Select (Drag and drop)	Why did you select?
	What will happen if you select sth else?
Observe	What is the value of?
	What is the purpose of?
Connect in circuit	How did you connect (series/parallel)?
	Why this type of connection?
Rotate	--
Double click	What is the value of specification xx?
	Can you change the value?
	Why did you change?
	What will happen if?
Right click	--
Construct/ Make	How did you construct?
	What type of analysis will you carry out?
	What function will xx circuit or system perform?
	Can it do yyy?
Display	What do you observe?
Reason out for selection	Why did you select the component/equipment/simulation property/plot/
Report observations	What is the nature of?
	State the type of.
Reason out for observation	Why did you get this output/result/plot?
Determine value of	How did you find out?
Explore relation between	What is the relation between the parameters?
Identify pattern	How are the parameters related?
Test a prediction	What will happen if?
Choose between	Which of the given xx will you choose for yyy function/purpose?
Reason out for the choice	Why did you choose xxx?
	What will happen if you choose yyy instead of xxx?

The following table gives the list of assessment questions aligned to the focus area/criteria you wish to fulfill through the virtual lab experiment.

Learning objective focus area	Assessment Questions
Practical Skills	1. State the equipment you used to construct the circuit. 2. State the purpose of each of the equipment used in the circuit 3. Which equipment you will choose to measure xxxx? 4. Why did you choose this equipment? 5. State the components you used to construct the circuit. 6. State the purpose of each of the components used in the circuit. 7. Which component you will choose for xxxx? 8. Why did you choose this component? 9. What will happen if x is chosen instead if y? 10. What type of input have you applied to the circuit? 11. State the specifications of applied input. 12. Why have you applied this input? 13. How will you observe the output? 14. How will you measure the output? 15. Write the value of the input parameter. 16. State the value of output parameter. 17. What will be the headings of your observation table? 18. How many readings have you taken? 19. Did you enter all the reading correctly? 20. How did you connect the equipment -Series/Parallel? 21. Why did you connect it in series/parallel? 22. What will happen if you reverse the connection? 23. What will happen if you reverse the component? 24. Is it possible to connect two or more components back to back? 25. Will you get the desired output if you connect the circuit with two or more components? 26. Observe the given circuit and state the types of connections.
Experimentation	1. State the equipment you used to construct the circuit. 2. State the purpose of each of the equipment used in the circuit. 3. Which equipment you will choose to measure xxxx? 4. Why did you choose this equipment? 5. State the components you used to construct the circuit. 6. State the purpose of each of the components used in the circuit. 7. Which component you will choose for xxxx? 8. Why did you choose this component? 9. What will happen if x is chosen instead if y? 10. What type of input have you applied to the circuit? 11. State the specifications of applied input. 12. Why have you applied this input? 13. How will you observe the output? 14. How will you measure the input? 15. How will you measure the output? 16. Write the value of the input parameter. 17. State the value of output parameter. 18. What will be the headings of your observation table? 19. How many readings have you taken? 20. Did you enter all the reading correctly? 21. How did you connect the equipment -Series/Parallel? 22. Why did you connect it in series/parallel? 23. What will happen if you reverse the connection? 24. What will happen if you reverse the component? 25. Is it possible to connect two or more components back to back? 26. Will you get the desired output if you connect the circuit with two or more components? 27. Observe the given circuit and state the types of connections. 28. What type of analysis will you carry out for the circuit? 29. Why would you carry out this analysis? 30. What simulation properties will you select for this analysis?

	31. Why would you choose these simulation properties? 32. What will happen if you choose yy analysis instead of xx? 33. What will happen if you choose yyy simulation properties instead of xxx? 34. If the circuit is modified what analysis will you carry out? 35. Why this type of analysis for the modified circuit? 36. What simulation properties will you choose for the modified analysis? 37. What parameter did you plot on the X/Y axes? 38. Why did you choose these parameters for the plots? 39. What is the nature of graph obtained? 40. What will happen if you plot x parameter on Y-axis and y on x-axis? 41. Can you plot x parameter on Y-axis and y on X-axis? 42. What is the scale selected for the two axes? 43. Why did you select these scales? 44. Can you change these scales?
Data Analysis	1. What type of analysis will you carry out for the circuit? 2. Why would you carry out this analysis? 3. What simulation properties will you select for this analysis? 4. Why would you choose these simulation properties? 5. What will happen if you choose yy analysis instead of xx? 6. What will happen if you choose yyy simulation properties instead of xxx? 7. If the circuit is modified what analysis will you carry out? 8. Why this type of analysis for the modified circuit? 9. What simulation properties will you choose for the modified analysis? 10. What parameter did you plot on the X/Y axes?

MARKDOWN

Markdown is a lightweight markup language with plain text formatting syntax. Its design allows it to be converted to many output formats, but the original tool by the same name only supports HTML.[8] Markdown is often used to format readme files, for writing messages in online discussion forums, and to create rich text using a plain text editor. Since the initial description of Markdown contained ambiguities and unanswered questions, the implementations that appeared over the years have subtle differences and many come with syntax extensions.

This is intended as a quick reference and showcase. For more complete info, see John Gruber's original spec and the Github-flavored Markdown info page.

This cheatsheet is specifically *Markdown Here's* version of Github-flavored Markdown. This differs slightly in styling and syntax from what Github uses, so what you see below might vary a little from what you get in a *Markdown Here* email, but it should be pretty close.

You can play around with Markdown on our live demo page.

(If you're not a Markdown Here user, check out the Markdown Cheatsheet that is not specific to MDH. But, really, you should also use Markdown Here, because it's awesome. http://markdown-here.com)

Table of Contents

Headers

```
# H1
## H2
### H3
#### H4
##### H5
###### H6

Alternatively, for H1 and H2, an underline-ish style:

Alt-H1
======

Alt-H2
------
```

H1

H2

H3

H4

H5

H6

Alternatively, for H1 and H2, an underline-ish style:

Alt-H1

Alt-H2

Emphasis

Emphasis, aka italics, with *asterisks* or _underscores_.

Strong emphasis, aka bold, with **asterisks** or __underscores__.

Combined emphasis with **asterisks and _underscores_**.

Strikethrough uses two tildes. ~~Scratch this.~~

Emphasis, aka italics, with *asterisks* or *underscores*.

Strong emphasis, aka bold, with asterisks or underscores.

Combined emphasis with asterisks and *underscores*.

Strikethrough uses two tildes. Scratch this.

Lists

1. First ordered list item
2. Another item
 * Unordered sub-list.
1. Actual numbers don't matter, just that it's a number
 1. Ordered sub-list
4. And another item.

Some text that should be aligned with the above item.

* Unordered list can use asterisks
- Or minuses
+ Or pluses

1. First ordered list item
2. Another item

- Unordered sub-list.

1. Actual numbers don't matter, just that it's a number
2. Ordered sub-list
3. And another item.
4. Some text that should be aligned with the above item.

- Unordered list can use asterisks
- Or minuses
- Or pluses

Links

There are two ways to create links.

```
[I'm an inline-style link](https://www.google.com)

[I'm a reference-style link][Arbitrary case-insensitive reference text]

[You can use numbers for reference-style link definitions][1]

Or leave it empty and use the [link text itself]

URLs and URLs in angle brackets will automatically get turned into links.
http://www.example.com or <http://www.example.com> and sometimes
example.com (but not on Github, for example).

Some text to show that the reference links can follow later.

[arbitrary case-insensitive reference text]: https://www.mozilla.org
[1]: http://slashdot.org
[link text itself]: http://www.reddit.com
```

[I'm an inline-style link](https://www.google.com)

[I'm a reference-style link](https://www.mozilla.org)

[You can use numbers for reference-style link definitions](http://slashdot.org)

Or leave it empty and use the [link text itself](http://www.reddit.com)

URLs and URLs in angle brackets will automatically get turned into links. http://www.example.com or http://www.example.com and sometimes example.com (but not on Github, for example).

Some text to show that the reference links can follow later.

Images

```
Here's our logo (hover to see the title text):

Inline-style:
![alt text](https://github.com/adam-p/markdown-here/raw/master/src/common/images/icon48.png "Logo Title Text 1")

Reference-style:
![alt text][logo]

[logo]: https://github.com/adam-p/markdown-here/raw/master/src/common/images/icon48.png "Logo Title Text 2"
```

Here's our logo (hover to see the title text):

Inline-style:

Reference-style:

Code and Syntax Highlighting

Code blocks are part of the Markdown spec, but syntax highlighting isn't. However, many renderers -- like Github's and *Markdown Here* -- support syntax highlighting. *Markdown Here* supports highlighting for dozens of languages (and not-really-languages, like diffs and HTTP headers); to see the complete list, and how to write the language names, see the [highlight.js demo page](http://softwaremaniacs.org/media/soft/highlight/test.html).

Inline `code` has `back-ticks around` it.

Inline code has back-ticks around it.

Blocks of code are either fenced by lines with three back-ticks ```, or are indented with four spaces. I recommend only using the fenced code blocks -- they're easier and only they support syntax highlighting.

```` 
```javascript
var s = "JavaScript syntax highlighting";
alert(s);
```

```python
s = "Python syntax highlighting"
print s
```

```
No language indicated, so no syntax highlighting.
But let's throw in a <b>tag</b>.
```
````

```javascript
var s = "JavaScript syntax highlighting";
alert(s);
```

```python
s = "Python syntax highlighting"
print s
```

```
No language indicated, so no syntax highlighting in Markdown Here (varies on Github).
But let's throw in a <b>tag</b>.
```

Again, to see what languages are available for highlighting, and how to write those language names, see the highlight.js demo page.

Tables

Tables aren't part of the core Markdown spec, but they are part of GFM and *Markdown Here*supports them. They are an easy way of adding tables to your email -- a task that would otherwise require copy-pasting from another application.

```
Colons can be used to align columns.

| Tables        | Are           | Cool  |
| ------------- |:-------------:| -----:|
| col 3 is      | right-aligned | $1600 |
| col 2 is      | centered      |   $12 |
| zebra stripes | are neat      |    $1 |

The outer pipes (|) are optional, and you don't need to make the raw Markdown line up prettily. You can also use inline Markdown.

Markdown | Less | Pretty
--- | --- | ---
*Still* | `renders` | **nicely**
1 | 2 | 3
```

Colons can be used to align columns.

Tables	Are	Cool
col 3 is	right-aligned	$1600
col 2 is	centered	$12

zebra stripes	are neat	$1

The outer pipes (|) are optional, and you don't need to make the raw Markdown line up prettily. You can also use inline Markdown.

Markdown	Less	Pretty
Still	renders	nicely
1	2	3

Blockquotes

> Blockquotes are very handy in email to emulate reply text.
> This line is part of the same quote.

Quote break.

> This is a very long line that will still be quoted properly when it wraps. Oh boy let's keep writing to make sure this is long enough to actually wrap for everyone. Oh, you can *put* **Markdown** into a blockquote.

Blockquotes are very handy in email to emulate reply text. This line is part of the same quote.

Quote break.

This is a very long line that will still be quoted properly when it wraps. Oh boy let's keep writing to make sure this is long enough to actually wrap for everyone. Oh, you can *put* Markdowninto a blockquote.

Inline HTML

You can also use raw HTML in your Markdown, and it'll mostly work pretty well.

<dl>
<dt>Definition list</dt>
<dd>Is something people use sometimes.</dd>

<dt>Markdown in HTML</dt>
<dd>Does *not* work **very** well. Use HTML tags.</dd>
</dl>

Definition list

Is something people use sometimes.

Markdown in HTML

Does *not* work **very** well. Use HTML *tags*.

Horizontal Rule

Three or more...

Hyphens

```
***

Asterisks

___

Underscores
```

Three or more...

Hyphens

Asterisks

Underscores

Line Breaks

My basic recommendation for learning how line breaks work is to experiment and discover -- hit <Enter> once (i.e., insert one newline), then hit it twice (i.e., insert two newlines), see what happens. You'll soon learn to get what you want. "Markdown Toggle" is your friend.

Here are some things to try out:

```
Here's a line for us to start with.

This line is separated from the one above by two newlines, so it will be a *separate paragraph*.

This line is also a separate paragraph, but...
This line is only separated by a single newline, so it's a separate line in the *same paragraph*.
```

Here's a line for us to start with.

This line is separated from the one above by two newlines, so it will be a *separate paragraph*.

This line is also begins a separate paragraph, but...

This line is only separated by a single newline, so it's a separate line in the *same paragraph*.

(Technical note: *Markdown Here* uses GFM line breaks, so there's no need to use MD's two-space line breaks.)

YouTube Videos

They can't be added directly but you can add an image with a link to the video like this:

```
<a href="http://www.youtube.com/watch?feature=player_embedded&v=YOUTUBE_VIDEO_ID_HERE
" target="_blank"><img src="http://img.youtube.com/vi/YOUTUBE_VIDEO_ID_HERE/0.jpg"
alt="IMAGE ALT TEXT HERE" width="240" height="180" border="10" /></a>
```

Or, in pure Markdown, but losing the image sizing and border:

```
[![IMAGE ALT TEXT HERE](http://img.youtube.com/vi/YOUTUBE_VIDEO_ID_HERE/0.jpg)]
(http://www.youtube.com/watch?v=YOUTUBE_VIDEO_ID_HERE)
```

TeX Mathematical Formulae

A full description of TeX math symbols is beyond the scope of this cheatsheet. Here's a [good reference](), and you can try stuff out on [CodeCogs](). You can also play with formulae in the Markdown Here options page.

Here are some examples to try out:

```
$-b \pm \sqrt{b^2 - 4ac} \over 2a$
$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}}}$
$\forall x \in X, \quad \exists y \leq \epsilon$
```

The beginning and ending dollar signs ($) are the delimiters for the TeX markup.

Equations in Markdown file

http://latex.codecogs.com/eqneditor/samples/example3.php

and just below the textbox select svg instead of gif/png
to embed in the markdown, simply copy the url "href........" and embed in the image tag in .md file.

html code :

image of the eqn

$\sum \prod \varepsilon \uplus \sum$"