# Hackathon: Detailed Content & Execution Plan [Owner: Kiran & Krishnan]

Hackathons are group activities for programmers to design ideas through collaboration and implement them fast together on premises. Usually, carried out over non-stop 24-48 hour windows working round the clock.

For now, we will adapt to the virtual-hackathon model which is individual skill development.

**Goals**

* To improve programming skills and unit testing skills
* To learn how to collaborate and brainstorm together on complex problems
* To learn how to plan, design and integrate different software components together within a stipulated period of time

The hackathon program at PUCSD consists of the following elements:

**Hackathon platform**

* Need to finalize a hackathon platform for delivering the hackathons in a sustainable way on a regular basis
* Selected platform needs to have the following features:
  + Ability to schedule in advance the start and end of a hackathon
  + Support for three distinct groups - organizers, participants and assessors
  + Automated verification of coding challenges

**Programming language/technology**

* Start with only 1 programming language to keep things simple
* Add more programming languages/technologies later on based on feedback and engagement levels
* How do we decide on the programming language ?
  + Go with Python as this is the most popular one and adds value to resume
  + Ask department faculty on what is the most commonly used programming language used by students

**Frequency & Duration**

* On weekends for 48 hours from 12:00 am Saturday to 11:59 pm Sunday
* Two factors need to be kept in mind while deciding the nature and content of the hackathons
  + We do not have first hand information about the technical competencies of the students
  + The initial set of assignments need to be at a reasonable level so that even average and below average students stay interested.
* The idea is to start with a set of individual programming problems and then, depending on the feedback and engagement levels, include group projects that are the norm in conventional hackathons.
* Frequency for individual programming assignments - Every weekend [too much - 150 people, excitement-factor]
* Frequency of group assignments - Once in a month culminating in a weekend
* Timeline
  + Start in August with individual assignments for 6-8 weeks
  + Start with group projects from mid-Sept/October

**Evaluation**

* Will consist of multiple checkpoints
* Checkpoint 1
  + Use automatic verification features of platform to check if the programs are producing the correct output
  + Submissions that pass this checkpoint are sent to the next checkpoint
* Checkpoint 2
  + Leverage our alumni for
    - Providing code review comments for working programs
    - Providing a live mentoring session to the authors of the working programs

**Pipe for hackathon content**

* Important to build a pipe of appropriate topics for the hackathon at the start so that it can be sustained
* Reach out to our alumni network for building this pipe
* Initial targets
  + Individual assignments: 15 assignments with increasing difficulty levels
  + Group projects:  6 projects

**Incentives**

* To sustain the interest levels in the hackathons, here are a few possible ways to incentivise the participants
  + Monetary rewards
  + LinkedIn recommendations/endorsement
  + Opportunity for top 3 programmers to present their design to the department
  + Certificates by departments

**Feedback**

* Optionally: to benefit a participant further some feedback on the code can be provided for the purpose of improvement

**Miscellaneous items/Mode of operation**

* Students will be expected to submit their assignments along with unit and system level tests
* Every hackathon challenge will be launched every Friday even via a teaser video from an alumni that describes the challenge and the expectations. The details of the challenge will only be available only on 12:00 am Saturday.
* Hackathon is open to students from all years (?) → after 2nd year (? level is different, keep first semester out.). Students are local residents - effort.
* Guidelines for the coaches on how to review.

**Challenges**

* Group projects may not be feasible until students return to the department as it may be quite challenging for students to work remotely on a group project.
* Funding required for using a hackathon platform
* Preventing students from copying code from the internet (oral viva ?)
* Feedback is useful but needs alumni time commitment and may overlap with the mentor program [don’t combine]. Should we allow them to speak to Alumni for advice while the hackathon is active.