Qiskit Hackathon Guide

Hi Qiskitters!

Welcome to the University Hackathon Planning Guide. As you start planning your campus or classroom hackathon, please reference this guide, which provides relevant resources and guidance as you plan a quantum computing hackathon at your university. We have included IBM-specific ideas and guidelines to help plan and host your event as successful (and pain-free) as possible.

Just as a reminder, in the world of quantum computing there are many different focuses. Engineering the physical hardware of a quantum computer is just as important as the software used to code your circuits. IBM Quantum strongly supports Qiskit, the open source SDK for running your code on real devices. Qiskit is built on Python and easy to install, which is why so many quantum computing hackathon end up focusing on Qiskit. However, there have been hackathons that are hardware design specific, machine-learning themed, and even game jams all using IBM Quantum tools.

Most things you'll find here are **suggestions**. The term "your event" is used on purpose. While IBM Quantum is more than happy to sponsor or support your event, at the end of the day this event is owned by you, your campus, and your group of collaborators. The IBM Quantum Community team is here to help and support you along the way. As a Qiskit-based event, there are a few rules that we'd like to see put into place at your event, and if your group is on board, the team can send swag, prizes, and even IBM Researchers and Qiskit Advocates as mentors or guest judges.

At the end of the day, you know your audience better than we do - that's why the IBM Quantum Community team defers to you for planning and logistics. If you disagree with something here, change it to suit your university or classroom. Just keep us in the loop!

If you only have a few minutes to read this document, you can skip to the TL;DR section at the end for the condensed overview. If you have any questions, comments, or concerns – please contact the IBM Quantum Community team at qiskit.events@us.ibm.com.

Good luck,

- The IBM Quantum Community Team

Overview

One good principle to always follow – the most successful events have a purpose. Are we just building random projects, or are we tackling some new papers published on arXiv? Maybe this hackathon is an end-of-semester event meant to show off the knowledge everyone learned. Or maybe it's a game jam where all teams are exploring the same concept and building programs to showcase it. No matter what your purpose or theme is, we recommend using that as a hook. It helps other participants get invested in the event and helps build excitement.

A second helpful principle – clearly state the roles between your team of organizers, the IBM team of supporters, and the participants at your event. You can think of it similar to a sports team. You and your organizers are the Head Coaches. You make the decisions on what, where, when, and who is joining. You have the final say over most aspects, but you take guidance from other sources. IBM Quantum helpers are the assistant coaches, here to help where you need it, but reluctant to step on your toes. They organize the IBMers to stop by, and deal with some of the smaller details that must be managed. Finally, we look at the participants as the athletes. At the end of the day, they're the entire reason for the event! Making sure they stay happy and engaged is crucial to the success of the entire team, and we are all working to achieve that goal.

Let's go team!

Code of Conduct

As IBM Quantum promotes and provides an inclusive and welcoming community, we appreciate everyone's support in providing and upholding our global code of conduct at your events. Having a strong code of conduct helps all the participants feel they're being treated fairly and have support in case an issue arises. IBM Quantum believes everyone deserves a fair playing field, and our code of conduct helps make that possible. The full IBM Quantum Community Code of Conduct is available on Github here.

Legal Agreement

IBM Quantum is interested in supporting your hackathon fully. In order to do this, an agreement will need to be secured between IBM and your group. Things like GDPR, international prize law, and personal information on registration lists need to be agreed upon to protect both parties from any legal repercussions. IBM cannot participate in a hackathon past an advisory role unless we sign an agreement or come to a similar understanding of responsibilities. If you are planning an event with IBM Quantum or Qiskit, and do not have an agreement in place, please reach out to Qiskit Events at qiskit.events@us.ibm.com at your earliest convenience.

Planning the Event

A vast majority of the work (and the issues) show up during the planning phase of a hackathon. Our best advice? **Plan with a team**. Having a group to rely on, delegate work to, and bounce ideas between becomes invaluable once you get into the nitty gritty details. In general, **you should not do everything yourself** unless you are really hoping to get overwhelmed. At minimum you should have one person focused on everything internal, and one focused on everything external.

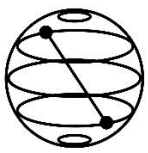
Digital, In-Person, and Hybrid

The very first thing you need to decide is what type of hackathon you want to run. Almost every decision after this will be influenced by your event being either digital, in-person, or a hybrid. Let's quickly look at the pros and cons of each

	Digital	In Person
Pros:	- Bigger pool of participants - Access to non-local mentors, judges, and speakers - Allows for people to participate while also taking care of responsibilities at home or with family, class, etc.	- Easiest control over communication, deadlines, and transitions between parts of the event - Networking and community building is easiest to facilitate - Team co-working simplified - Minimize tech issues - Distribution of swag & prizes becomes seamless
Cons:	- Contest Law and GDPR compliance - Communication is easier to be missed - Higher drop-off rates of participants both before and during the event - May need to pay for platform access - Possibility of tech issues rises	- Space can be limited, or expensive to rent - Limited to only those who can be away from family, pets, etc. for long periods of time - Food should be provided by organizers

There's no perfect answer to what you should plan. Some people opt to combine both approaches and run a hybrid hackathon. You can imagine that hybrid event strengthens some aspects and makes others (like communication) even more difficult.

We recommend you **pick whatever your participants will be most comfortable doing**. That's what the success of your event will ultimately rely on, so catering to your group of hackers and helping them feel ready will help you in the long run.



Map out your Timeline

To start planning, you should map out the different anticipated beats of the event in advance. These are things like workshops, when registration will open, social events, when the hackathon begins and ends, and more. At the beginning stages, these dates should be penciled in --- not permanent. Things can change fast!

Most hackathons we've worked with have included the following actions on planned dates:

- Announcement of the event via social media
- Announcement of the schedule (that's what you're planning now!)
- A public or private "Intro to Qiskit" workshop for your registered attendees
- A social event, game night, or something similar
- A kickoff ceremony to officially start the hackathon, possibly a keynote speaker
- An "end of hacking" announcement, Zoom call, or ceremony
- Structured time for teams to present their projects to the judges
- Awards ceremony

The above sessions form the skeleton of most hackathons. You can always add more sessions on top of these — maybe a company highlight, a more in-depth tutorial, or Q&A with researchers in the field.

If you're considering **not** doing one or more of the bullet points above, it's good to discuss with your co-organizers why that decision was made. There might be a really obvious reason why, but making sure to communicate all decisions will help the event go smoothly.

Work with the Community

Many people have organized Qiskit hackathons at this point. Ask your friends or classmates if they've had experience putting a hackathon together. Many of the Qiskit Advocates, the network of the strongest Qiskit Community members around the world, have taken part in or planned hackathons. You can use their website to find Advocates near you and reach out to them through Slack. You might also get some responses from the #meetups channel. Finally, check in with the other clubs or organizations on your campus. More often than not, there will be overlap of students who want to participate.

Who to Invite

Having a mix of the right skills and personalities makes all the difference for a successful hackathon. To reach a variety of participants, you can ask the heads of local physics, programming, engineering, or other related groups around campus to invite a certain number of their members or co-promote the hackathon. Promoting the event over Twitter, Discord, and other social media can also yield big results. You can try to identify students in your classes or clubs who have a strong passion for building things. Not sure which clubs to reach out to first? Try looking for local chapters of IEEE, ACM, Computer Science, or Data Science clubs. Some undergrad quantum computing clubs have begun making partnerships with similar clubs at other universities, look for any of those at other schools in your area.

To get an even broader selection of participants from your university or the schools nearby, you can set up an open signup form in something like Airtable. If using an open signup, you should be careful to include information you might need in the future to help determine if someone is eligible. Remember **it's illegal in the US to collect any information on anyone under 13 years old**, and different platforms have their own rules around use. Stating a minimum age of at least 16 is recommended.

You also may want to check the location of participants. Whether you're going with in-person or digital, this is a good idea. If you're planning everything in your local time zone, having people sign up from the other side of the world could cause issues, especially with team cohesion. This info also might be useful in case you're planning on mailing anything.

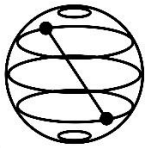
The design of your registration form can be straightforward, and the fields below can be a great starting point for creating one of your own:

Name	Email	Location	Strongest Skills	Age	School
Kit Kiss	Example@abc.com	Storrs, CT	Programming, leadership	27	UConn

Again, the fields you choose to use or not to use are up to your team. When it comes to what focus area backgrounds you might want at your event, here are some groups that tend to find quantum computing interesting:

Programmers, Physicists, Chemists, Artists, Computer Scientists, and Tech Enthusiasts

To further expand your pool of participants, it might be helpful to include both graduate and undergrad students. Professors can, and should, act as mentors or support members during your hackathon. However, **professors should not be contestants in the hackathon.** This can create additional legal requirements and difficulties, as well as make it seem "unfair" for other teams.



The more participants you can handle, the stronger your event will be. For universities**, 40 – 80 participants is a great range** for your hackathons success. Anything less than 25 starts to get a little thin, and anything over 100 gets overwhelming for only a few judges to sort through.

Announcing your hackathon 3 - 5 weeks before your event is a solid amount of time to put everything else into place, and give everyone enough of a heads up to plan their participation. Registration should finish at least 3 days before the hackathon begins. This ensures that you have an accurate head count for food, social events, and swag. Sending reminders the week of, and the day before your event begins, is also a great idea!

Suggested Pre-Hackathon Events

Hosting pre-hackathon events is extremely useful for everyone across the board. It helps keep the participants engaged and excited; it gives you as organizers a clear roadmap and milestones to hit; and it helps ensure everyone knows what they're doing.

One aspect to get out of the way early is for every participant to register through the IBM Quantum Experience and start playing around with the circuit composer. Hackathon participants need to have an IBMid created in order to participate, so this is a good early activity.

Maybe the most important workshop to hold is a Qiskit 101 tutorial. Installing and running Qiskit for the first time may be a little confusing. Having someone on site to troubleshoot and walk through a few of the Qiskit tutorials hosted on the public GitHub is a good introduction.

The Coding with Qiskit YouTube series, and our open source Qiskit Textbook are both popular resources that people have used to help get people up to speed. At a recent Madrid hackathon, the organizers analyzed the skill sets of their registered participants, and then arranged four separate meetups prior to the event to get everyone comfortable and coding with Qiskit at a high level.

Other topics you can cover in your pre-hackathon sessions are:

- How to use Qiskit Pulse
- Overview of Grover's Algorithm
- Building and playing quantum games
- How to simulate quantum errors
- Different applications of the VQE algorithm

Rules

Qiskit Hackathons around the world usually follow the same general rules. We suggest that you include these in your hackathon.

- You must use Qiskit in some way in your project
- Maximum number of members per team: 5 people
- Projects must be uploaded to a public repository by the end of hacking, using an open-source license
- Teams must show off their project in some public way (elevator pitches, science fair, etc.)

While the rules are never the most exciting part of an event, it's important for all contestants to know the judging is fair. Any coaches or advisors present at the hackathon should be working with all teams; not just one. Often, a coach is also a final judge. Setting clear rules and expectations makes the entire event stronger.

Running the Event

Agenda

The agenda is the most malleable piece of a Hackathon. Some organizers opt for a 24-hour overnight hackathon, while others do a small 8-hour sprint. Online hackathons can last for 1 or 2 weeks, giving participants time to work on their projects in between their normal day.

Look at your participants and decide what makes the most sense for them. The happiness and comfort of your participants is crucial. The most common in-person hackathon lasts two days. The most common online hackathons last 7 days. One the following page find two sample agendas of real hackathons we've seen in the past.

Sample Agenda 1 – (In Person)

Day 1

8am - 9am	Breakfast	Venice Ballroom
9am - 10am	Introduction, Rules, Example projects	Conference Room 4
10am - 11am	Qiskit Refresher	Conference Room 4
11am - 10pm	Hacking time!	with your group

Day 2

9am - 12pm	Finish hacking	with group
12pm	deadline for project to be published	Github
12:15 pm	Lunch, networking	Venice Ballroom
2pm - 4pm	Presentation to the Judges	Conference Room 1
4:30 am	Awards Ceremony	Venice Ballroom
7:30 pm	Social Event	Huskies Bar & Grill

Sample Agenda 2 – (Online)

Participant Registration	
Qiskit Workshop 1	June 19 7pm
Qiskit Workshop 2	June 22 7pm
Form Teams and Pick a Project	June 22 - June 24
Qiskit Workshop 3	June 24 7pm
Officially submit your team's project idea	June 25 By 10am
Begin building your project (Hacking Begins!)	June 25 12pm
Mentor Check-in	June 28 Various Times
Online Game Night	June 29 10 pm
Project Submission Deadline (Hacking Ends)	July 1 12pm
Judging Begins	July 1
Social Hour on Gather.town	July 1 5pm
Awards Ceremony	July 2

Good hackathons can look extremely different. **The key is making sure all participants understand what they need to do, and where they need to be**. There's nothing worse than kicking off your awards ceremony and only having 1/3 of your participants because most of them forgot to look at their email for the link. **A clear agenda, sent to participants early (and often), will be extremely helpful.**

Getting your agenda ironed out as early as you can will be helpful for your other decisions as well. It's easy to slightly modify the schedule as things come up, but starting the process with clear goals and benchmarks in mind will help everyone. This also keeps the organizing team focused on what they're each responsible for with deadlines attached.

Knowing the skill level of your audience will be essential to deciding how to intersperse the learning options throughout your event. At the end of the day, hackathons are meant to be fun events that help students learn quantum computing and use Qiskit. If your audience just finished a Quantum Information Science course on campus, you don't need as many tutorials as, say, a group of interested students with no physics background.

Tech Support & Tough Questions

It's rare to find a hackathon where not a single participant has some type of issue. They can come in various shapes and sizes. *(One time we hosted an in-person hackathon where we overloaded the bandwidth of our site, and had to purchase emergency wifi hotspots, but that's a story for another time...)*

Being responsive, flexible, and empathetic will help you solve almost every problem.

Here are a few situations we've seen happen multiple times:

- One team has been diligently working all night, and right when they needed to submit their project, their wifi crashed. What do you do?
- A group misread the time zones on the schedule and showed up an hour late for the workshop. Now they feel unprepared for the hackathon and are considering dropping out. How do you help them?
- A team submitted their project an hour early, but discovered a bug just before judging begins (and past the submission deadline). They ask you if they're allowed to fix it. Are they?
- A team's demo worked just fine earlier, but during the presentation to the judges is breaks and they're unable to get it to work. Can they show the judges again after they get it running?

There's no universal answer to any of these questions. You and your team should **communicate about what issues come up, and what you're doing to resolve it**. All teams want the event to feel fair, and that they have an equal shot at winning. That's why it's also important to be transparent about any modifications you're making for certain teams.

Use logic and empathy when solving these issues. Maybe you let a team fix their bug, but deduct a point from their score. Or, maybe you let them fix the bug, but let all other teams know they may fix 1 bug also. Or, maybe you do not allow them to change anything. It's up to you and your team to make the choice.

Most importantly: **Be consistent with the choices you make for all teams.**

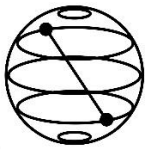
Another good thing to keep in mind is to **have Physics and Computer Science helpers** on hand to help problem solve for smaller issues during the event. Some participants may struggle with the Python, while others need help on the core concepts of quantum physics. Having people available to give quick answers allows the event to continue smoothly.

Group Formation

Ok, it's the morning of your big event. A huge group of students sit in your auditorium (or Zoom call) listening to your welcome speech. Everyone is excited, brimming with ideas, waiting to get started. The only question is, how *do* you get started?

You'll find some people show up ready with an idea they want to build, and others come just looking to have fun and make something cool. **We recommend a 60-second pitch session** to get things started. This can be very informal – standing in a big circle, sitting on the stage with a microphone, or just shouting to a huddled mass of students. Each person with an idea should take 60 seconds to describe the idea they have, and what kind of skills would be needed to build it. Someone pitching a new optimization tool for Qiskit may need a different group than someone pitching a quantum Raspberry Pi project.

It's very common to see people a little shy at the beginning of a pitch session, and even into the team formation part. If possible, you should **have 5 - 10 ideas premade by coaches or organizers.** They can start off the pitches and make people feel more comfortable to share their own idea.



After the pitches are done, give members 10 minutes to organize into teams to work with for the remainder of the hackathon. Use this time as organizers to help anyone looking lost. Find out what they're passionate about and help them find a team to join. Remember – team size should be a maximum of 5 participants.

In a digital world, this process is much more difficult. It's really easy for someone feeling nervous or lost to just log off and walk away. **We see this very often with digital events.** Think of what tools would be best to use to organize something like this. Maybe a shared Google Sheet where people put their own names under a project idea. Or maybe you host the event in gather.town and allow people to form groups that way. You could also list project ideas as Issues on a Github repo, and have participants add their names directly.

If you're worried about a lack of pitches or project ideas, there's a lot of places to turn to. We recommend this blog post and this YouTube video for inspiration.

Food

Depending on how you've arranged your schedule or what you've planned, you may need to provide food to your groups. As part of signing the legal agreement, IBM may be able to send an appropriate amount of money to cover food costs. A good idea is to ask each participant during the signup process to list any dietary restrictions. Pizza is always a hit, but eating healthy is fun too 😊

How to pick a winner

At a university hackathon, the way prizes are awarded should always be consistent. Although IBM may send guest coaches, they are **not the final judges.** IBM coaches can provide thoughts and insight, but the final choice needs to always be someone from your local planning team or university. If you don't know who to ask, course TAs, professors, and club presidents are a good place to start.

It is very important to share the judging categories with the contestants, so they know exactly what to build towards. It's also imperative for the judges to know this before the event starts, so they know what to keep an eye out for.

Here is our guidance for how Judges should evaluate each project:

- **Originality and Uniqueness (25%)**
  - Compared to what you've seen before, how unique is this project? How interesting do you find it? Did the team attempt something new or difficult?
- **Usefulness and Complexity (25%)**
  - Will other people be able to use this project? Was the project thoughtful in how it was designed? How functional is the project as of judging?
- **Quantum Community Benefit (25%)**
  - Will this project help the community at large? Can people use this for research or further develop it? Will this project help others learn and understand quantum computing?
- **Presentation (25%)**
  - Did the team represent their project well? Was the team able to explain why they made certain decisions? Did the entire team get a chance to speak?

Finally, it’s best if judges reserve their actual “judging” until the end of the hackathon. If a team struggles for the first 2 hours, that shouldn’t be a detriment to their final product.

Prizes, Awards, and Wrap

IBM Quantum can provide small swag for all participants, and special prizes for the winners as well. It's important to celebrate the work put in for such an event. Having certificates of completion, or some other way for participants to show their accomplishment is also strongly suggested.

For some participants, especially game designers, the feedback and critique is just as important as winning a prize is. Please make sure you **provide time for groups to chat with judges afterwards** and get this feedback. Growing the community is a key priority with the IBM Quantum Community Team, and constructive feedback is key to that mission's success.

Most organizers feature a post-hackathon event, sometimes at a local restaurant or (age-permitting) bar as a social networking event. Some of the best connections happen after the event, because of the shared experience between participants. Wherever you decide to host yours, be safe!

Finally, **please share your event pictures with us!** IBM Quantum (and the entire community!) loves to see a good hackathon in action. Thank you, and good luck!

TL;DR

Not everyone has time to read everything. Here are the most important pieces of advice, ordered by the strongest recommended actions at the top.

- Host a few pre-hackathon events aimed at teaching fundamentals of quantum and qiskit. Distribute the Juypter notebooks or learning materials to everyone present.
- When dealing with issues that teams face, be consistent with all your decisions. Other teams will notice!
- Clearly communicate roles and responsibilities to the entire organizing team, multiple times during planning and the day-of.
- Send out the schedule to attendees ASAP.
- Social events help give people a break from learning and problem solving. They are your friend. They're especially helpful post-event to let coaches and participants have a longer dialogue about what worked, what didn't, and how to improve.
- Allow teams to form organically after participants pitch their ideas.
- Share the judging criteria with teams before the hackathon starts.
- Do not identify who the final judges will be until the end of the event. All organizers, judges, and coaches should act the same during the hackathon: mentoring teams and helping them solve issues
- Have a mixed crowd. Programmers, Physicists, Chemists, Artists, Computer Scientists, and Tech Enthusiasts are all great additions to a hackathon
- Make sure your venue has strong WiFi
- Have fun!