CT101 Computing Systems Hackathon

Theme 2026: "IoT is Bold" and "Old is Gold"

Computing Systems (CT101)

University of Galway

Dr. Waqar Shahid Qureshi

School of Computer Science

What is the Hackathon?

A Collaborative Learning Experience

- **Worth:** 15% of module grade
- **Duration:** 12 weeks (4 phases)
- **Team Size:** 2-3 students
- **Choose Your Track:**
  - **IoT is Bold** - Raspberry Pi
  - **Old is Gold** - Legacy Servers

Real-World Skills

- Hardware/Software Integration
- Team Collaboration
- Project Management
- System Design
- Problem Solving

Hackathon Timeline Overview

Phase	Week	Deliverable	Weight
Phase 1	Week 1-4	Project Abstract (1 page)	10%
Phase 2	Week 6-7	Functional Requirements (4-5 pages)	25%
Phase 3	Week 10	**Live Presentation + Demo**	**35%**
Phase 4	Week 14	Design Document (8-12 pages)	30%

Hardware Checkout: Mid-Week 8

Hardware Return: End of Week 12

Two Themes, Two Technologies

Theme 1: "IoT is Bold"

Hardware: Raspberry Pi 4 with Sense HAT
Focus: Embedded systems, sensors, real-time data

Theme 2: "Old is Gold"

Hardware: Legacy PC Desktop Systems
Focus: Web servers, databases, system architecture

Choose ONE theme - limited hardware available (max 15 teams per theme)

Theme 1: "IoT is Bold"

Raspberry Pi & Sense HAT

https://www.youtube.com/watch?v=kZwtn-iazm0

What is Raspberry Pi?

A Credit Card-Sized Computer

Perfect for IoT Projects

- ✅ Low power consumption
- ✅ Small form factor
- ✅ Real-time sensor processing
- ✅ Network connectivity
- ✅ Camera support
- ✅ Rich software ecosystem

Raspberry Pi Hardware Specifications

Available Kit Components:

Component	Specification	Purpose
Raspberry Pi 4	Quad-core ARM Cortex-A72	Main processing unit
Sense HAT	8×8 LED Matrix + Sensors	Input/Output interface
Camera Module v3	12MP, 1080p video	Visual monitoring
Power Supply	USB-C, 5V/3A	Power delivery
HDMI Cable	Micro HDMI to HDMI	Display connection

Total Kits Available: 10-15 systems

Raspberry Pi Sense HAT: The Brain

Originally for Space!

Developed for International Space Station's Astro Pi program

Built-in Sensors:

- Temperature sensor
- Humidity sensor
- Barometric pressure
- Accelerometer (motion)
- Gyroscope (orientation)
- Magnetometer (compass)

Output Capabilities:

- 8×8 RGB LED Matrix
- 5-button mini joystick
- Visual feedback display

Documentation:

- Sense HAT Docs
- Getting Started Guide

IoT is Bold: Project Categories

1. Environmental Monitoring & Surveillance

- Monitor temperature, humidity, pressure
- Capture photos when conditions change
- Create time-lapse sequences
- Detect disturbances with accelerometer

2. Motion Detection & Interaction

- Use accelerometer for tilt sensing
- Orientation-based animations

IoT is Bold: Project Categories (cont.)

3. Interactive Display & Gaming

- Tilt-controlled games (marble maze, balance)
- Joystick navigation

4. Data Logging & Analysis

- Collect sensor readings over time
- Create visualizations and graphs

IoT is Bold: Why Choose This Theme?

Best for students who:

- Enjoy hands-on hardware projects
- Like working with sensors and physical computing
- Want to learn embedded systems
- Are comfortable with Python programming
- Interested in real-time data processing
- Want to build interactive systems

Theme 2: "Old is Gold"

Legacy Server Systems & Infrastructure

What is a Server System?

A Computer That Serves Others

- Provides services to clients
- Runs 24/7 (high availability)
- Handles multiple users
- Stores and processes data
- Manages resources
- Different types for different needs

Why "Old is Gold"?

- Learn fundamental architecture
- Understand server types
- Database management
- Web technologies
- System optimization
- Real-world deployment

Legacy PC Server Hardware

Available Equipment:

Component	Specification	Purpose
Desktop PC	Various CPU/RAM configs	Server platform
Storage	HDD/SSD available	Data persistence
Network	Ethernet connection	Client connectivity
Power	Standard AC power	Continuous operation

Total Systems Available: 20 PCs

Flexible Network Infrastructure: On request

Seven Server Types to Choose From

1 Static Web Server

Serves pre-built files

- HTML, CSS, JavaScript
- Same content for all users
- Fast and simple
- Example: Cohort websites

2 Dynamic Web Server

Generates pages on-demand

- Database-driven content
- User-specific data
- Example: Assignment submission system

3 Media Streaming Server

Delivers audio/video

- Multi-user streaming
- Format conversion
- Access control

4 Search/Document Server

Manages document collections

- Full-text search
- Advanced filtering
- Rapid discovery

Seven Server Types (continued)

5 Game Server

Manages multiplayer sessions

- Real-time player sync
- Matchmaking
- Game state management
- Persistent progression

6 Application Server

Executes business logic

- Complex workflows
- Transaction management
- High concurrency
- Example: Student Information System

7 Chat Server

Real-time messaging

- WebSocket connections
- Multi-client synchronization
- Instant message delivery
- Example: Campus Community Chat

Server Type 1: Static Web Server

Purpose: Serve pre-built HTML, CSS, JavaScript files

How it Works:

1. Files stored on disk
2. Content identical for all users

Technology Stack:

- Apache HTTP Server / Nginx
- HTML5, CSS3, JavaScript
- File system storage

Example Project: Academic year cohort websites (CS Year 1, Year 2, etc.)

Server Type 2: Dynamic Web Server

Purpose: Generate pages in real-time based on database queries

How it Works:

1. Client requests data
2. Server generates HTML dynamically
3. Different users see different content
4. Content changes based on user/context

Technology Stack:

- Node.js/Express, Flask, Django
- PostgreSQL/MySQL database
- Server-side templating

Example Project: Student Assignment Submission & Grading System

Server Type 3: Media Streaming Server

Purpose: Stream audio/video to multiple users simultaneously

How it Works:

1. Store media files on server
2. Client requests stream
3. Server adapts quality to bandwidth
4. Multiple simultaneous connections

Technology Stack:

- FFmpeg for transcoding
- HLS/DASH protocols
- WebRTC for real-time
- Storage management

Server Type 4: Search/Document Server

Purpose: Provide searchable access to document collections

How it Works:

1. Index documents for fast searching
2. Client submits search query
3. Server performs full-text search
4. Results ranked by relevance
5. Advanced filtering and facets

Technology Stack:

- Elasticsearch / Apache Solr
- Database for metadata
- Web frontend for search interface

Server Type 5: Game Server

Purpose: Manage multiplayer game sessions with real-time sync

How it Works:

1. Players connect to server
2. Server authenticates players
3. Matchmaking creates game sessions
4. Server synchronizes game state
5. Persistent player data storage

Technology Stack:

- WebSocket for real-time
- Game loop architecture
- Player state management
- Database for persistence

Server Type 6: Application Server

Purpose: Execute complex business logic and multi-step workflows

How it Works:

1. Handle complex transactions
2. Coordinate multiple operations
3. Integrate with databases
4. Manage concurrent users

Technology Stack:

- Java/Spring, .NET, or Node.js
- Relational database (PostgreSQL)
- Transaction management

Example Project: Simplified Student Information Management System

Server Type 7: Chat Server

Purpose: Manage real-time message exchange between clients

How it Works:

1. Clients establish WebSocket connections
2. Messages sent to server
3. Server broadcasts to all connected clients
4. Instant delivery (no polling needed)

Technology Stack:

- Socket.IO / WebSocket
- Node.js/Express backend
- Database for message history
- Web frontend for chat UI

Example Project: Campus Community Chat Platform

Old is Gold: Web Technologies Foundation

Resources:

- HTML Tutorial
- CSS Tutorial
- JavaScript Tutorial
- PostgreSQL Tutorial
- Socket.IO Chat Tutorial

Old is Gold: Why Choose This Theme?

Best for students who:

- Enjoy web development and databases
- Like building software systems
- Want to learn server architecture
- Are interested in backend development
- Prefer software over hardware
- Want to learn system design patterns

Hardware Availability & Constraints

Limited Resources!

Theme	Hardware Available	Max Teams	Team Size
IoT is Bold	10-15 Raspberry Pi kits	**15 teams MAX**	2-3 students
Old is Gold	20 PC systems	**15 teams MAX**	2-3 students

First come, first served - Register your theme in Week 1!
Cannot change themes after Week 5 (except special circumstances)

Assessment Breakdown

Total Hackathon Weight: 15% of module grade

Phase	Deliverable	Weight	Individual/Team
Phase 1	Project Abstract (1 page)	10%	Team
Phase 2	Functional Requirements (4-5 pages)	25%	Team
Phase 3	**Live Presentation + Demo**	**35%**	**Individual**
Phase 4	Design Document + Reflection	30%	Individual

Individual assessment - Each team member evaluated separately!

Phase 3: The Hackathon Event (Week 10)

3-Hour Live Showcase

Time	Activity
0:00-0:30	Setup, tech check, final integration
0:30-3:50	Team presentations (8 min each)
	• 5 minutes presentation + demo
	• 3 minutes Q&A from judges
3:50-4:00	Judging deliberation
4:00+	Prize distribution & celebration

Your demo must be functional and ready!

Judging Criteria (Phase 3 - 35%)

Criterion	Weight	What Judges Look For
Demonstration Quality	40%	Fully functional, handles issues smoothly
Presentation Clarity	30%	Clear narrative, good pacing, engaging
Technical Communication	20%	Explains approach well, judges understand
Time Management	10%	Fits 8-minute window perfectly

Score Scale: 1-5 points per criterion

Hardware Checkout Process

Timeline:

1. **Mid-Week 8:** Hardware checkout begins
2. **Weeks 8-12:** Development period with hardware
3. **End of Week 12: MUST RETURN** hardware

Responsibilities:

- Sign checkout form
- Designated hardware owner per team
- Responsible for safekeeping
- Inspection upon return

Damage Policy:

- Normal wear: No charge
- Repairable damage: To be assessed
- **Report damage within 24 hours!**

Start Without Hardware (Weeks 1-7)

IoT is Bold Students:

Use Simulators!

- Sense HAT emulator
- Online Raspberry Pi simulator
- Test code before hardware
- Plan GPIO connections

Resources:

- Raspberry Pi Desktop
- Sense HAT emulator included

Old is Gold Students:

Use Your Own Computer!

- Install server software
- Set up database locally
- Develop web interface
- Test functionality

You can:

- Use your laptop/desktop
- Set up virtual machine
- Use cloud development
- Test before deployment

Prizes & Recognition

Three Prize Categories:

1. **Best Bold and Gold** (Overall Winner)

2. **Best Bold Implementation**

3. **Best Gold Implementation**

Generative AI Policy

When You CANNOT Use AI:

❌ **Phase 1: NO AI for brainstorming, problem exploration**
❌ **Phase 2: NO AI for technical writing, document generation**
❌ **Phase 4 Reflection: NO AI for analysis, reflection sections**

When You CAN Use AI:

✅ **Phase 3+: Code generation, debugging (with attribution)**
✅ **Phase 4 Writing: Sentence improvement (must document)**

Why? Develop authentic thinking, problem-solving, and communication skills!

Key Deadlines (Don't Miss These!)

Week	Deadline	What's Due
Week 1	End of week	Team formation decision
Week 4	End of week	Project Abstract (1 page)
Week 6-7	Mid-week	Functional Requirements (4-5 pages)
Week 8	Mid-week	Hardware checkout
Week 10	TBA	**HACKATHON EVENT**
Week 12	End of week	Hardware return
Week 14	End of week	Final Design Document

Getting Started: Your Action Items

This Week (Week 1):

1. ☐ Find 1-2 teammates (form teams of 2-3)
2. ☐ Decide: IoT is Bold vs. Old is Gold
3. ☐ Choose team name
4. ☐ Assign initial roles
5. ☐ Create GitHub repository
6. ☐ Start brainstorming project ideas

By Week 4:

7. ☐ Submit Project Abstract (1 page, 500 words max)
8. ☐ Secure your hardware allocation

Questions to Ask Yourself

Choosing IoT is Bold?

- Do I like hardware projects?
- Am I comfortable with Python?
- Do I want to work with sensors?
- Do I like physical computing?
- Can I handle hardware debugging?

Choosing Old is Gold?

- Do I prefer software development?
- Am I interested in databases?
- Do I like web technologies?
- Do I want to learn server architecture?
- Am I comfortable with multiple languages?

Both themes are equally valuable - Choose what excites YOU!

Resources and Documentation

IoT is Bold:

- Raspberry Pi Documentation
- Sense HAT Docs
- Picamera2 Manual
- Getting Started Projects

Old is Gold:

- PostgreSQL Tutorial
- HTML/CSS/JS
- Socket.IO Chat
- Real-time Chat Tutorial

GitHub Repository Structure

```
your-project/
├── README.md                 # Project overview, setup guide
├── requirements.txt          # Python dependencies
├── src/                      # Source code
│   ├── main.py              # Main application
│   ├── sensors.py           # Sensor modules (IoT)
│   └── server.js            # Server code (Old is Gold)
├── docs/                     # Design documents, diagrams
│   ├── abstract.pdf
│   ├── requirements.pdf
│   └── design.pdf
├── hardware-setup/           # GPIO diagrams (IoT only)
├── data/                     # Sample data, logs
└── CHANGELOG.md              # Version history, progress
```

FAQ: Common Questions

Q: Can I change teams after Week 1?

A: Yes, until Week 6. After Week 6-8, requires approval. After Week 8: NO changes.

Q: What if a teammate stops participating?

A: (1) Address directly (2) GTA intervention (3) Instructor intervention. Team continues with reduced scope.

Q: Can I use my own Raspberry Pi?

A: You may use your own Pi 4, but must check out official Sense HAT from school.

Q: What if my hardware breaks?

A: Report within 24 hours with photos. Normal wear covered. Damage assessed case-by-case.

FAQ: More Questions

Q: Can everyone choose IoT is Bold?

A: NO - Limited to 15 teams due to hardware constraints.

Q: What happens if my abstract is rejected?

A: Rare! You'll get feedback:

- Approved (proceed)
- Approved with feedback (incorporate, no resubmit)
- Revision requested (48 hours)
- Rejected (choose different project, 3 days)

Q: How is my individual grade determined?

A: Work distribution docs (Phases 1, 2), individual presentation performance (Phase 3), individual reflection (Phase 4).

Success Tips

To Excel in This Hackathon:

1. Start Early - Don't wait for hardware!
2. Communicate - Regular team meetings
3. Document Everything - GitHub is your friend
4. Test Incrementally - Small steps, frequent testing
5. Ask for Help - GTAs and instructor are here
6. Practice Presenting - Rehearse your demo
7. Manage Time - Use project management tools
8. Stay Organized - Keep track of deadlines

What Makes a Great Hackathon Project?

✅ **Clear Problem Definition** - What are you solving?
✅ **Appropriate Scope** - Achievable in timeframe
✅ **Technical Depth** - Shows learning and skill
✅ **Working Demo** - Functional is better than ambitious but broken
✅ **Good Documentation** - Clear, complete, professional
✅ **Team Collaboration** - Everyone contributes meaningfully
✅ **Innovation** - Creative approach or unique solution
✅ **Polished Presentation** - Practice makes perfect!

Remember: It's About Learning!

This Hackathon is Designed to Help You:

- Apply theoretical knowledge to practical projects
- Learn new technologies through hands-on experience
- Develop teamwork and collaboration skills
- Practice technical communication
- Build portfolio projects for your career
- Experience the full software/hardware development lifecycle
- Gain confidence in your technical abilities

The goal is learning, not perfection!

Good Luck!

University of Galway
School of Computer Science