

An Internet of Things Introductory Curriculum for Grades 9-12 Computer Science Classes

Pramod Abichandani, Deepan Lobo, Prateek Shekhar

Abstract—The emergence of the Internet of Things (IoT) has had a transformative effect on our society and has inspired educators to develop innovative approaches to educate the next generation of Computer Science (CS) professionals. This paper presents the design and development of an introductory IoT course suitable for grades 9-12 Computer Science classes. Information about the course content, intended outcomes, and evaluation techniques are presented. The course was introduced in 2 high schools in the US. The course includes a capstone project where the students identified a real-world problem and developed an IoT-based solution to address it. Formative and summative technical evaluation results are presented and suggest that the course provided an effective learning experience for students. The information presented here provides guiding principles for developing an IoT-based curriculum geared towards K-12 education while also exposing the students to CS fundamental.

I. INTRODUCTION

Internet-of-Things (IoT) is a disruptive technology which has enabled ubiquitous and pervasive computing scenarios [1]. The IoT refers to the system of inter-connected computing devices, mechanical and digital devices, and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. The IoT market is continually growing and are expected to grow to 1.6 trillion U.S. dollars in size by 2025 [2]. There is an increasing interest in using IoT technologies in agriculture, food processing industry, environmental monitoring, security surveillance, robotics and drones, education, and other significant industries [3]–[5]. These industries correlate closely with current rapidly growing need for software developers equipped with skills of computer programming that leverage the intersection between hardware and software. Researchers have recognized this need to train future software developers equipped with the knowledge in IoT.

One crucial skill of a software developer is effective computer programming. Characteristics that define a skilled computer programmers are assessed on four dimensions namely critical thinking, problem-solving, computational thinking, and system design [6]. The need for computational thinking and improvements in the programming skills of individuals continue to receive widespread attention to better prepare future citizens [7]. Today, basic knowledge of programming is part of almost all engineering programs curricula. However, usually, both students and instructors confronted with several problems during these courses, and the programming concepts and language syntax may become barriers for learning programming and an inhibiting factor for motivation among students [8]. Researchers report that high number of concepts and skills to learn in a limited time, absence of incorporation of

interactive media and instant feedback instruction, and lack of mathematical background are the main barriers in learning programming [9].

IoT presents a novel paradigm for enhancing computer science education. An IoT system involves physical computing at its core – the use of inexpensive microcomputers that run optimized code to collect, analyze, and share data with other devices or a cloud server. The inherent integration between hardware, software, real-time data, processes, and human interaction, IoT has the potential to serve as an excellent platform for teaching computer science topics in an experiential format. IoT can be used to introduce C.S. principles in K-12 due to its heavy focus on physical computing, and the availability of easy-to-use programming languages such as Python and NodeMCU/Lua [10]. These modern programming languages reduce the barrier to entry in C.S. for 9-12 students. For example, the authors in [11] report on the use of IoT hardware and software for 6-12 summer programs. Students focused on creating IoT prototypes using pre-made code templates. Similarly, recent projects from the U.S. National Science Foundation (NSF) are also creating an IoT pedagogical ecosystem for integrated C.S. and Software Engineering education for 9-12 students [12]. Architecturally, IoT-based curricula can leverage a learning framework consisting of a collection of integrated computing devices and instruments for mapping abstract computer programming concepts and theories into practical, hands-on experiential knowledge [13].

This paper presents the design and development of an IoT course that introduces computer programming concepts for grades 9-12. The coursework is designed around the instructive-constructive-active-passive (ICAP) cognitive modes of engagement presented in the ICAP framework [14]. The curriculum was tested at 2 high schools located in the US. The key contributions are highlighted as follows:

- A course design that introduces computer science concepts to students of grades 9 - 12 through the lens of IOT.
- Quantitative formative and summative evaluation results of the first iteration of this course that provide insights about the effectiveness of this course.

II. METHODOLOGY

A. 4-Layer Model

The multi-dimensional nature of IoT technologies have been succinctly captured in multi-layered conceptual architectures by several authors [4], [15]–[19]. A popular model is the IoT World Forum Reference Model that has been widely adopted

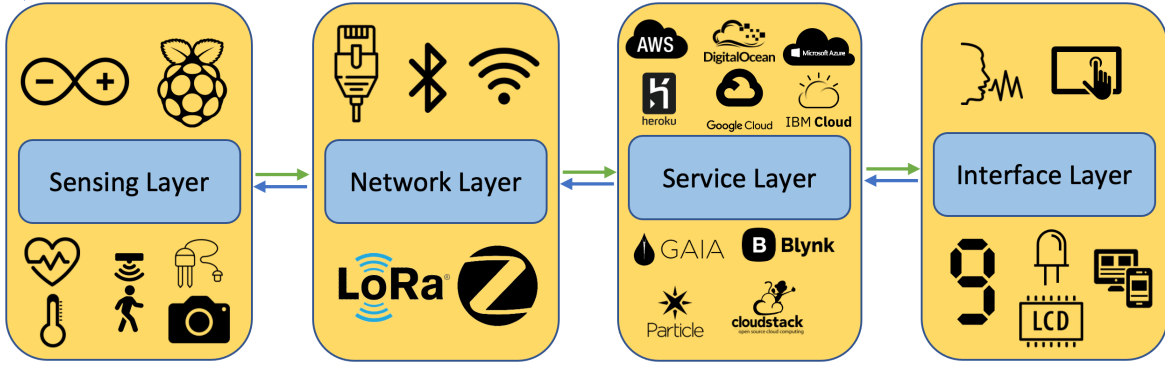


Fig. 1: The IoT 4-layer architecture encompassing IoT systems [6].

in the literature [20]. While these conceptual architectures feature different numbers and names for IoT layers, they have strong overlaps in terms of the core IoT paradigms. In its most general form, IoT can be conceptually expressed as a system formed by the interaction of 4 key dimensions: real-world sensing, communication networks, software services, and user interfaces. These dimensions result in a 4-layer model that was proposed by *Da et. al.* in [4], as depicted in Figure 1. When viewed through the lens of IoT/STEM education, this 4-layer model allows for educators to design curriculum, pedagogy, and assessment in a multi-disciplinary manner while immersing students in the technical and societal challenges associated with IoT [6].

B. Curricular Modules

The accessibility of single-board computing devices such as Arduino and Raspberry Pi, and the abundance of online resources has lowered the barrier of entry for students interested in learning programming. Thus, one of the main motivations of this study is to present an organized course structure that leverages the experiential component of IoT devices with the software concepts of programming. The curriculum is designed around programming IoT sensor hardware with a modern computer programming language. Figure 2 depicts the conceptual diagram of the essential tasks in the coding exercises.

1) **Weeks 1 - 2: Introducing Python:** The programming language used in the curriculum is Python3, which is an interpreted, high-level and general-purpose programming language that is concise and easy to read. Python is considered as one of the most popular programming languages and combined with its accessibility and versatility makes it the programming language of choice for first time users [21].

This module guides the students through the software installation process of the Python programming environment, code editor, and necessary software driver support for the IoT sensor hardware. Students were introduced to several Python Integrated Development Environments (IDEs). IDEs contain many useful features to write, edit and debug code, visualize and inspect data, store variables, and present results. While IDEs might differ visually, the programming language does not. Thus, the learning curve is limited to understanding Python's syntax.

After this introductory setup, students were introduced to fundamental programming constructs of Python. These programming constructs include variables, mathematical operations, input/output commands, conditionals, loops, and lists. These programming constructs introduced in the first two weeks of the course provide the building blocks for developing the first Python program to interface with sensor hardware.

2) **Weeks 3-5: Sensing Layer Integration:** During the next weeks, students were introduced to IoT sensor hardware. The sensing layer is the primary layer in any IoT-based system and includes sensors or transducers embedded in real-world environments [6]. The sensors included in the curriculum are classified into the two categories namely environmental sensors, and proximity/motion sensors. A key aspect of the curriculum was the use of easy-to-use Python Application Programming Interface (API) developed by the authors. These APIs enabled easy interface with the sensor hardware and actuators. Each lecture session began with the instructor first providing background about the hardware. Students were then asked to discuss the real-world applications of the sensor and identify any commercial products that use the particular sensor. This discussion was followed by a hands-on session where the students were introduced to the template Python code needed to interface with the hardware. After the students successfully interfaced with the hardware they moved on to associated coding exercises. These exercises allowed students to explore different logical/programming constructs associated with the sensors.

3) **Weeks 6-7: Interface Layer Integration:** This module was covered in tandem with the sensing layer and guided the students to create interfaces that allowed users to interact with an IoT system. This layer includes hardware displays such as LEDs (mono-colored or multi-colored), Liquid Crystal Displays (LCDs), touch screens, and actuating devices such as motors and buzzers. The students learned how to use Python programs to create real-time visualizations of the sensor data obtained from the sensing layer. Software interfaces such as dashboards, alerts, prompts, and notifications form a strong focus in this layer.

4) **Week 8: Network Layer Concepts:** The network layer module covered theoretical aspects of technologies that enable connection between IoT system elements and exchange information with each other. This layer is significant as IoT-

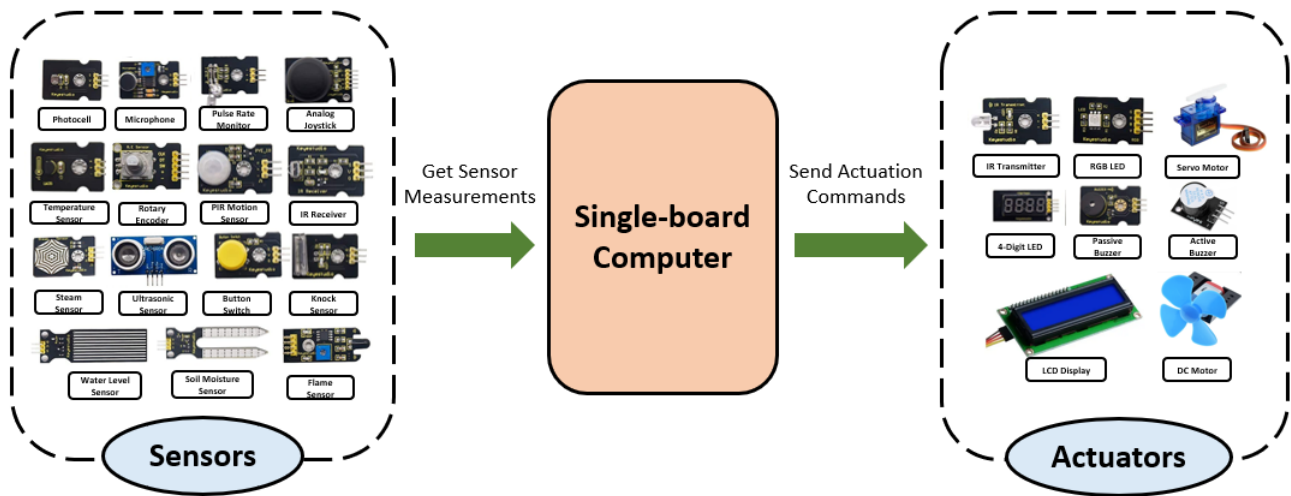


Fig. 2: Conceptual diagram of essential tasks in coding exercises

based systems are connected through wired or wireless networks. Wi-Fi, Bluetooth, Zigbee, LoRa are the most popular wireless technologies for communication used by IoT devices and systems. The instructor covered the basic overview of these technologies in class while students discussed their experiences with these technologies. Students also discussed the strengths and weaknesses of these technologies.

5) **Week 9: Service (Cloud) Layer Concepts:** The final module introduced concepts of cloud technologies and services. Cloud services provide a backbone to the internet of things and is a crucial component in understanding the working of internet of things. This module covered theoretical concepts on how cloud services function followed by discussions in the classroom. Students were introduced to different vendors (Amazon, Google, Microsoft, and IBM among others) that provide cloud services and the breadth of cloud services provided by such vendors.

The assignments contained a variety of coding exercises involving the sensors and actuators. Through these programming exercises, students' ability to leverage the template Python code provided in the introduction and build sophisticated functionalities was evaluated. The coding exercises were designed according to the 3A and 3B standards (grade level 9 -12) outlined by the Computer Science Teachers Association (CSTA) [22]. Rubrics to formatively evaluate the student's comprehension of the CSTA standards-aligned concepts were used. These concepts included 1) Algorithms and Programming, 2) Computing Systems, 3) Data and Analysis, 4) Impacts of Computing, and 5) Network and the Internet .

C. Final Project

The final component of the course featured student teams working on a capstone IoT project. The project spanned 6 weeks of the term and began after the students completed their assignments. The primary instructional objective of this project was to allow student teams to work together effectively and apply the skills/knowledge learned from the course to build a tangible solution to a real-world problem. Students were

free to decide their project topic and project domains varied from environmental monitoring, home automation, connected infrastructure, and connected biomedical devices. The project was broken down into the following phases:

1) **Weeks 5 - 6: Team Formation and Project Proposal:**

Leveraging their newly acquired knowledge and skills, students worked together and brainstormed ideas to solve a real-world problem through an IoT solution. Students were asked to form teams and propose their projects by Week 7 of the terms. Student teams could have between 2 and 3 members. Students were also free to work independently.

A template of the proposal document was provided to the students. The template included prompts to engage students to think deeply about their project idea and team composition. The prompts included questions about them: 1) problem statement, 2) commercial/societal significance, 3) technical approach, 4) key innovation, 5) expected outcome and deliverables, and 6) resources needed.

Students used their school's Learning Management Systems (LMS) to float ideas, discuss potential collaborations, and coalesce around a coherent project narrative with other classmates. During this period, students were required to engage with the instructors through weekly 1-on-1 meetings during their office hours to craft their projects' scope and technical deliverables appropriately.

2) **Weeks 7 - 9: Weekly In-Class Project Progress Presentations:**

After the project proposal was approved by the instructors, students were expected to lead the project execution while having short interactions with the instructors. The key requirement during this phase was a 3-minute long weekly in-class progress presentation that outlined their weekly progress. The progress presentation featured three slides that covered the weekly progress, challenges, and expected milestones for next week.

3) **Week 10: Project Video:**

The final deliverable for the

TABLE I: Rubric for IoT Capstone Project

#	Metric	Excellent	Average	Poor
1	Project Completion	All deliverables met	Some deliverables met	No deliverables met.
2	Video presentation	Meets all the video presentation requirements, excellent presentation skills	Partly meets the requirements of the video presentation, And mediocre presentation skills	Does not meet any of the requirements of the video presentation, or no video submitted.
3	Communication	Excellent at communicating challenges and progress made	Average at communicating challenges and progress made	Poor at communicating challenges faced and little to no progress weekly
4	Innovation	Implemented a unique feature or aspect to the project	Proposed an innovative concept but did not deliver	No innovation implemented
5	Realizability	Project is working, and real-world application demonstrated	Project is working, but real-world applicability, not possible	Project is not working, and real-world applicability is not possible.
6	Report organization	Excellent presentation demonstrates understanding of IoT technology and programming	Average presentation, satisfactory understanding of IoT technology and programming	Poor organization of report, and sub par understanding of IoT technology and programming

capstone project was a presentation along with a technical report. The presentation was to include the following information: 1) Introduction and project statement, 2) Demonstration of the functions of the IoT project, 3) Description of design challenges and approaches adopted to solve them, 4) Learning resources referred to while working on the project, 5) Anticipated challenges at the beginning of the project, and 6) Approaches taken to solve the challenges.

III. TECHNICAL EVALUATION

Formative assessments were used for the evaluation of Python and Sensors module of the curriculum. Summative assessments were used to evaluate the final project.

A. Direct Formative Assessment

The course featured five assignments that were due every week. The assignments consisted of a list of mini-projects featuring sensors and actuators being interfaced using Python. Each assignment was designed around the list of sensor hardware discussed in class that week. The mini-projects were formulated such that the students were met with an increasing level of difficulty, where the specifications for each project built on top of the previous project. The assignments were worth 10 points each. A detailed description of the assignments are provided as follows:

- 1) Basic components: This assignment featured the basic components of an embedded systems such as LEDs, RGB LED, push button, and analog sensors including the photo resistor, and thermistor. Students were guided through the steps of creating simple circuits on a breadboard. The program specifications introduced the students to essential Python programming concepts such as data types, print statements, and conditional logic.

- 2) Loops: This assignment introduced students to the concept of looping in programming. Looping is a beneficial feature of programming which allows the code to be written once and repeated as many times as needed, thus shortening the size of the program. The questions included hardware from the first assignment and required the students to reinforce programming concepts such as printing, and conditional logic.
- 3) Actuators: This assignment introduced a variety of actuators used in embedded systems such as DC motor, servo motor, and active/passive buzzers. Students were guided through the process of reading sensor measurements obtained from the environment designing logic to create real-world actuation in a meaningful way. For example, increasing the speed of DC motor based on increasing temperature measurements to simulate a cooling system.
- 4) Displays: This assignment introduced sophisticated display technologies found in real-world systems such as 7-segment LED, LCD, and Dot-Matrix LED. Creating expressive prompts, and warning messages to convey to users important information is a crucial component of the interface layer of an IoT system. This assignment also introduced additional sensor hardware to expand the available input devices for the project, namely water level sensor and a Passive Infrared (PIR) sensor.
- 5) Advanced sensors: Finally the students were introduced to a range of advanced sensors which add significant value to real-world IoT systems. This module covers the infrared transmitter/receiver, accelerometer and gyroscope sensor, real-time clock, analog joystick, and keypad. The exercises in this assignment also brought in some of the hardware covered in earlier assignments to showcase the versatility of the Python programs the students have developed thus far.
- 6) Cloud Services: The students are guided through the steps of setting up the service layer for their IoT system through the use of WebSockets and Amazon Web Services (AWS). The exercises in this assignment had the students recreate the IoT systems developed in earlier exercises by integrating them with a web-app.

TABLE II: Descriptive Statistics of grades obtained by 19 students on their IoT-based Python programming assignments (A1 - A5). Each assignment and quiz was worth 10 points.

Ass	A1	A2	A3	A4	A5
Mean	8.47	7.18	8.58	9.42	8.89
Std. Dev.	2.52	1.52	2.13	2.20	1.41

Table II depicts the mean and standard deviation of each assignment grade for 19 students. The mean of the grades

received by the students for the IoT-based Python assignments varied from the 7.18 to 9.42. The standard deviation observed for the IoT-based Python assignments varied from 1.52, to 2.52.

B. Direct Summative Assessment

The project submission was evaluated according to the rubric provided in Table I. The rubric was adopted from resources available on [23] while keeping in mind the objectives of the capstone project. Based on the final presentations, 90% demonstrated a fully working project as per the proposed description, and only 10% presented a partially working project where they had a physical model, but the system was not working as intended.

IV. CONCLUSION

This paper presented an introductory IoT-based, standards-aligned curriculum for CS classes for grades 9-12. The focus of the curriculum was on the use of IoT architectural layers to create hands-on, physical computing-centric learning experience. The final projects allowed students to explore team-based IoT solutions for real-world problems. Based on the formative and summative evaluations obtained during the first implementation of this curriculum, it was found that students were able to grasp IoT and CS concepts effectively. Future work will focus on refining the curriculum and its implementation, and collection of qualitative data via surveys and focus groups. As IoT-centric education becomes more pervasive in high schools, the curricular information provided in this paper may serve as a guide for educators and researchers.

REFERENCES

- [1] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking, "A review on Internet of Things (IoT), Internet of everything (IoE) and Internet of nano things (IoNT)," in *2015 Internet Technologies and Applications (ITA)*. IEEE, 2015, pp. 219–224.
- [2] L. S. Vailshery, "Forecast end-user spending on IoT solutions worldwide from 2017 to 2025," <https://www.statista.com/statistics/976313/global-iot-market-size/>, 2021.
- [3] C. Iaboni, H. Patel, D. Lobo, J.-W. Choi, and P. Abichandani, "Event camera based real-time detection and tracking of indoor ground robots," *IEEE Access*, vol. 9, pp. 166 588–166 602, 2021.
- [4] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.
- [5] S. Li, L. Da Xu, and S. Zhao, "5g Internet of Things: A survey," *Journal of Industrial Information Integration*, vol. 10, pp. 1–9, 2018.
- [6] P. Abichandani, V. Sivakumar, D. Lobo, C. Iaboni, and P. Shekhar, "Internet-of-things curriculum, pedagogy, and assessment for stem education: A review of literature," *IEEE Access*, 2022.
- [7] D. Pérez-Marín, R. Hijón-Neira, A. Bacelo, and C. Pizarro, "Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children?" *Computers in Human Behavior*, vol. 105, p. 105849, 2020.
- [8] D. Topalli and N. E. Cagiltay, "Improving programming skills in engineering education through problem-based game projects with scratch," *Computers & Education*, vol. 120, pp. 64–74, 2018.
- [9] M. Rizvi, T. Humphries, D. Major, M. Jones, and H. Lauzun, "A cs0 course using scratch," *Journal of Computing Sciences in Colleges*, vol. 26, no. 3, pp. 19–27, 2011.
- [10] "NodeMCU an open-source firmware based on ESP8266 Wi-Fi." [Online]. Available: https://www.nodemcu.com/index_en.html
- [11] A. Jaklič, "Tot as an Introduction to Computer Science and Engineering: A case for NodeMCU in STEM-C education," in *2020 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2020, pp. 91–95.
- [12] "National Science Foundation(NSF)-Division of Research on Learning." [Online]. Available: https://www.nsf.gov/awardsearch/showAward?AWD_ID=2010259
- [13] J. He, D. C.-T. Lo, Y. Xie, and J. Lartigue, "Integrating internet of things (iot) into stem undergraduate education: Case study of a modern technology infused courseware for embedded system course," in *2016 IEEE frontiers in education conference (FIE)*. IEEE, 2016, pp. 1–9.
- [14] M. T. Chi and R. Wylie, "The icap framework: Linking cognitive engagement to active learning outcomes," *Educational psychologist*, vol. 49, no. 4, pp. 219–243, 2014.
- [15] J. Gómez, J. F. Huete, O. Hoyos, L. Perez, and D. Grigori, "Interaction system based on Internet of Things as support for education," *Procedia Computer Science*, vol. 21, pp. 132–139, 2013.
- [16] S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.
- [17] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys and tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [18] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future generation computer systems*, vol. 56, pp. 684–700, 2016.
- [19] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [20] "IoT World Forum Reference Model," http://cdn.iotwf.com/resources/72/IoT_Reference_Model_04_June_2014.pdf, (last visited 2021-07-26).
- [21] F. Klint, "Python Is More Popular Than Ever." [Online]. Available: <https://www.wired.com/story/python-language-more-popular-than-ever/>
- [22] D. Seehorn, S. Carey, B. Fuschetto, I. Lee, D. Moix, D. O'Grady-Cunniff, B. B. Owens, C. Stephenson, and A. Verno, *CSTA K–12 Computer Science Standards: Revised 2011*. ACM, 2011.
- [23] "Project based learning works," Buck Institute for Education, Last accessed on 05-20-2021. [Online]. Available: <https://my.pblworks.org/>