POLYTECHNIC UNIVERSITY OF THE PHILIPPINES
College of Engineering
Computer Engineering Department
Mabini Campus, Sta. Mesa

CMPE 101
Computer Engineering as a Discipline

ENGR. ROLITO L. MAHAGUAY
ENGR. PEDRITO M. TENERIFE, JR.

COMPUTER ENGINEERING AS A DISCIPLINE

CMPE 101

Engr. Rolito L. Mahaguay, *MSE-CpE, PCpE*

Published and distributed by:

Polytechnic University of the Philippines

Address	: 1016 Anonas St., Brgy. 630, Sampaloc, Sta. Mesa, Manila
Website	: https://www.pup.edu.ph
Email Address	**:** inquire@pup.edu.ph
Tel. No.	**: (+63 2) 5335-1PUP (5335-1787) or 5335-1777**

THE VMPGO

VISION

PUP: A Leading Comprehensive Polytechnic University in Asia

MISSION

Advance An Inclusive, Equitable, and Globally Relevant Polytechnic Education towards National Development.

PHILOSOPHY

As a state university, the Polytechnic University of the Philippines believes that:

Education is an instrument for the development of the citizenry and for the enhancement of nation-building; and,

That meaningful growth and transformation of the country are best achieved in an atmosphere of brotherhood, peace, freedom, justice and nationalist-oriented education imbued with the spirit of humanist internationalism.

SHARED VALUES AND PRINCIPLES

1. Integrity and Accountability
2. Nationalism
3. Sense of Service
4. Passion for Learning and Innovation
5. Inclusivity
6. Respect for Human Rights and the Environment
7. Excellence
8. Democracy

GOALS OF THE COLLEGE/CAMPUS

1. To empower transformational leaders to form social bonding and encourage voluntary work behavior; to create a culture of innovation and continuous improvement; to build a diverse, transparent, inclusive workforce; and reduce the organization's environmental impact

2. To offer curricula that are relevant and responsive to the changing needs of the industry and society; the ability of curriculum developers to translate knowledge about new development into curriculum content and structure; to promote critical thinking, a sense of adventure, and an openness to adapt challenges of their future workplace and give them a hierarchical system for grades levels/subjects within aims and objectives for individual lessons.
3. To increase students' attention, and focus, promote a meaningful learning experience, encourage higher levels of student performance, motivate students to practice higher-order thinking skills; to prepare students to become productive, creative, innovative and dynamic in their chosen fields of specialization and to provide state of the art facilities of learning to optimize student development; tap potentials of students, faculty, administrative staff, and other stakeholders in formulating policies for institutional development.
4. To prepare holistic approaches to inculcate appropriate values that are necessary to build a humane, disciplined, nationalist, and independent society and to develop student's physical, emotional, social, and intellectual well-being through physical, emotional, social, and intellectual well-being through providing opportunities for students to learn and grow in all areas of their lives; to create a supportive inclusive environment where students feel safe and respected to be active participants in their learning for students to connect with others, build relationships and to help students develop a sense of purpose and direction.
5. To build a culture of trust, deliver honest feedback, foster open communication, delegate responsibilities and tasks, and support growth opportunities to empower faculty members and employees. Also, to increase productivity and innovation; improve morale and satisfaction: better decision-making; increase engagement with students and clients and make empowerment part of our university organization's culture and vision.
6. To a renowned leader and center of excellence in product utilization research, feasibility study, development, and technology transfer; develop the culture of collaborative research among students, faculty, and employees; to partner with industry and other research institutions in strengthening research capabilities of faculty, employees, and students to facilitate presentation of research outputs in international for a, their publication in recognized local and international journals; and to develop the culture of collaborative research among students, faculty, and employees.

7. To contribute to the attainment of Vision, Mission, Goal, and Objectives (VMGO) distinctively include complying with the rules and policies of the Polytechnic University of the Philippines (PUP); striving for academic excellence, participating actively in universities activities, becoming a role model, passing the board exam and conducting research. To maintain and enhance its high academic standards in the performance of its functions of instructions, research, and adaptive community for extension.

8. To create value for each company and leverage combined expertise by offering students internship partnerships through a Memorandum of Agreement (MOA); undertake outreach and research-based extension programs by tapping all stakeholders' expertise and other resources.

9. To increase understanding of stakeholders needs and expectations, improve communication and collaboration, and involve all stakeholders in enhancing student, faculty, and employee development programs, build trust and rapport with stakeholders, and get input from stakeholders on critical decisions.

10. To ensure that our curricula possess Social Development Goals (SDG) such as social equity, justice, diversity, inclusion, democratic participation, empowerment, end poverty, to protect the earth's environment and climate and to ensure that students, educators, and stakeholders can enjoy peace and prosperity; to provide training to students that will enable them to become potent instruments for socio-economic development, produce technologies for commercialization or livelihood improvement, and achieve long-term economic growth.

PROGRAM DESCRIPTION

Computer Engineering is a four-year degree program that deals with the study of computer systems. The curriculum covers both software and hardware and develops the student's ability to analyze computer systems, designs, construction of electronic equipment and its peripherals. Since computer science is directed to the theory and technology of computation, the curriculum does not specialize along traditional lines that divide hardware and software, systems and applications, or theory and experiment. Rather, a unified approach to the design and analysis of computers and of computing structures is employed. This background prepares the student for placements as computer engineers in government industry. It also qualifies them for related job with computer manufacturers and consisting of firms as systems programmers as well as application programmers with scientific, research, and business organizations. The ethical considerations with respect to the profession is an important component of the program of study.

The BSCpE curriculum has four (4) tracks of specialization namely: (a) Computer Network Engineering; (b) Machine Learning; (c) Big Data; and (d) System Development. It is designed to prepare graduates in accordance with the institutional and program outcomes.

The curriculum has a total of 188 credit units comprising of 129 units of technical courses. These technical courses include 12 units of mathematics, 8 units of natural/physical sciences, 4 units of basic engineering sciences, 11 units of allied courses, 78 units of professional courses (common), 12 units of professional courses (specialized), and 4 units on the job training (OJT).

The non-technical courses in accordance with CMO 20 s. 2013 - The New General Education Curriculum consist of 59 units of general education courses distributed as follows: 33 units of core courses, 9 units of GEC electives, and 3 units of Life and Works of Rizal.

INSTITUTIONAL LEARNING OUTCOMES (ILOS)
A PUP graduate imbibed:

1. **Creative and Critical Thinking**
   Graduates use their imaginative as well as rational thinking ability to life situations to push boundaries, realize possibilities, and deepen their interdisciplinary and general understanding.

2. **Effective Communication**
   Graduates are proficient in the four macro skills in communication (reading, writing, listening, and speaking) and can use these skills in solving problems. Making decisions, and articulating thoughts when engaging with people in various circumstances.

3. **Strong Service Orientation**
   Graduates exemplify the potentialities of an efficient, well-rounded and responsible professional deeply committed to service excellence.

4. **High Level of Leadership and Organizational Skills**
   Graduates are developed to become the best professionals in their respective disciplines by manifesting the appropriate skills and leaderships qualities.

5. **Community Engagement**
   Graduates exemplify the potentialities of an efficient, well-rounded and responsible professional deeply committed to service excellence.

6. **Effective Communication**
   Graduates are proficient in the four macro skills in communication

7. **Adeptness in the Responsibility Use of Technology**
   Graduates demonstrate optimized use of digital learning abilities, including technical and numeracy skills.

8. **Sense of Personal and Professional Ethics**
   Graduates show desirable attitudes and behavior either in their personal and professional circumstances.

9. **Passion to Life-long Learning**
   Graduates are enabled perform and function in the society by taking responsibility in their quest to know more about the world through lifelong learning.

10. **Sense of National and Global Responsiveness**
    Graduate's deep sense of national compliments the need to live in a global village where one's culture and other people culture are respected.

PROGRAM LEARNING OUTCOMES (PLOS)
By the time of graduation, the students of the program shall have the ability to:

1. Apply knowledge of mathematics and sciences to solve complex engineering problems.
2. Design and conduct experiments as well as analyze and interpret data.
3. An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental. social political, ethical, health and safety, manufacturability, and sustainability, in accordance with standards.

4. Function on multidisciplinary teams.
5. Identify, formulate, and solve complex engineering problems.
6. Understanding of professional and ethical responsibility.
7. Communicate effectively.
8. Understand the impact of engineering solutions in a global, economic, environmental and societal context.
9. Recognize the need for, and an ability to engage in life-long learning.
10. Recognize and assess contemporary issues.
11. Use techniques, skills, and modern engineering tools necessary for engineering practice; and
12. Understand engineering and management principles as a member and leader in a team, to manage projects and in multidisciplinary environments.

COURSE LEARNING OUTCOMES (CLOS)

1. Comprehend key networking principles, architectures, and protocols, including the OSI and TCP/IP models.
2. Design, configure, and manage wired and wireless networks, utilizing devices like routers, switches, firewalls, and applying IP addressing and subnetting.
3. Diagnose and resolve network issues using troubleshooting tools and methodologies, while optimizing performance through monitoring and load balancing
4. Apply security best practices, including firewalls, VPNs, access controls, and encryption, to protect network infrastructures
5. Analyze and incorporate emerging technologies like SDN, IoT, and network automation to enhance network scalability, management, and performance.

PREFACE

Computer Engineering as a Discipline course is a comprehensive introductory course designed to explore the core principles and evolving areas within the field of computer engineering. Over 18 weeks, students will engage in 1-hour weekly sessions covering a wide range of topics, from fundamental concepts such as software development and hardware design to emerging technologies like artificial intelligence, IoT, and cloud computing. The course aims to provide a solid foundation for understanding the interdisciplinary nature of computer engineering, emphasizing both hardware and software aspects, as well as their integration in real-world applications.

Through lectures, discussions, and hands-on activities, students will also explore ethical and legal considerations in engineering, professional certifications, and technopreneurship. By the end of the course, they will be well-prepared for further study, research, or capstone projects in advanced areas of computer engineering, and ready to tackle the challenges and opportunities in this ever-evolving field.

COURSE CONTENT

MODULE 1

Introduction to Computer Engineering

Learning Objectives:

- Understand the fundamental concepts of computer architecture.
- Identify the major hardware components in a computer system.

Introduction to Computer Engineering

I. **Question that you might ask about Computer Engineering Professions**

- The Computer Engineering degree encompasses a wide range of topics, including operating systems, computer architecture, computer networks, robotics, artificial intelligence, and computer-aided design.
- It is a program designed to meet the rapidly expanding demand for engineers with strong design skills.
- Computer Engineering is a four-year degree program that deals with the study of computer systems.
- The curriculum covers both software and hardware and develops the student's ability to analyze computer systems, designs, construction of electronic equipment and its peripherals.

Computer Science and Engineering Trend

Average Salary by Discipline, Bachelor's Degrees In USA

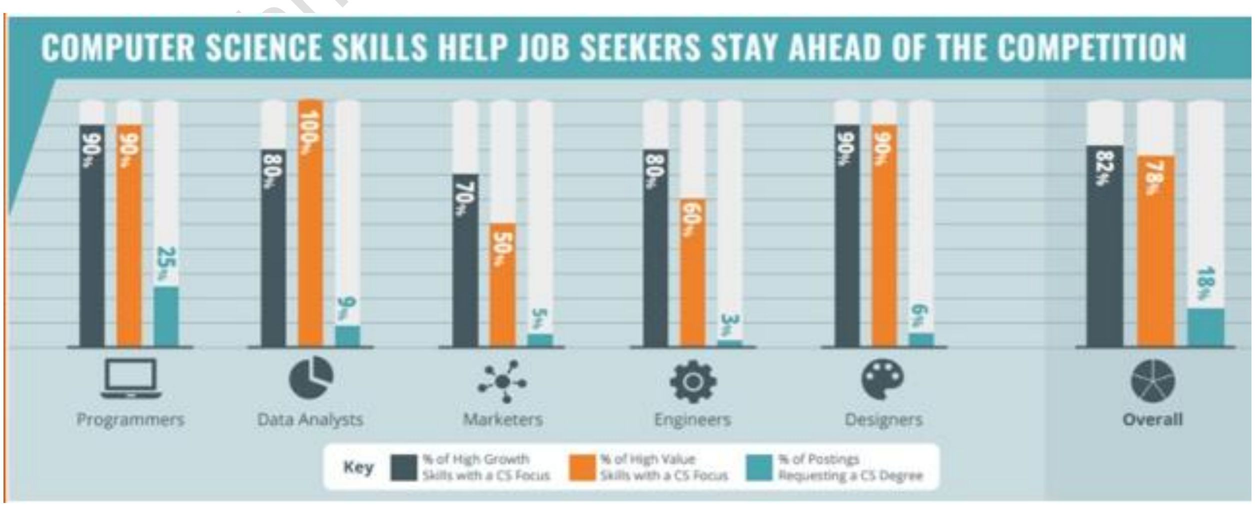
Discipline	2019 Salary	2018 Salary	Percent Change
Engineering	$69,188	$66,521	4.00%
Computer Science	$67,539	$66,005	2.30%
Math & Sciences	$62,177	$61,867	0.50%
Business	$57,657	$56,720	1.70%
Social Sciences	$57,310	$56,689	1.10%
Humanities	$56,651	$56,688	-0.10%
Agriculture & Natural Resources	$55,750	$53,565	4.10%
Communications	$52,056	$51,448	1.20%

Source - National Association Of Colleges And Employers 2019 USA

Computer Science Major	Degree	Entry Level	Mid - Career
Computer Science (CS) & Engineering	Bachelor's	$74,000	$120,100
Computer Science (CS)	Bachelor's	$68,800	$113,900
Information & Computer Science	Bachelor's	$81,400	$104,600
Computer Science & Information Technology	Bachelor's	$61,400	$97,700
Applied Computer Science	Bachelor's	$55,300	$92,000

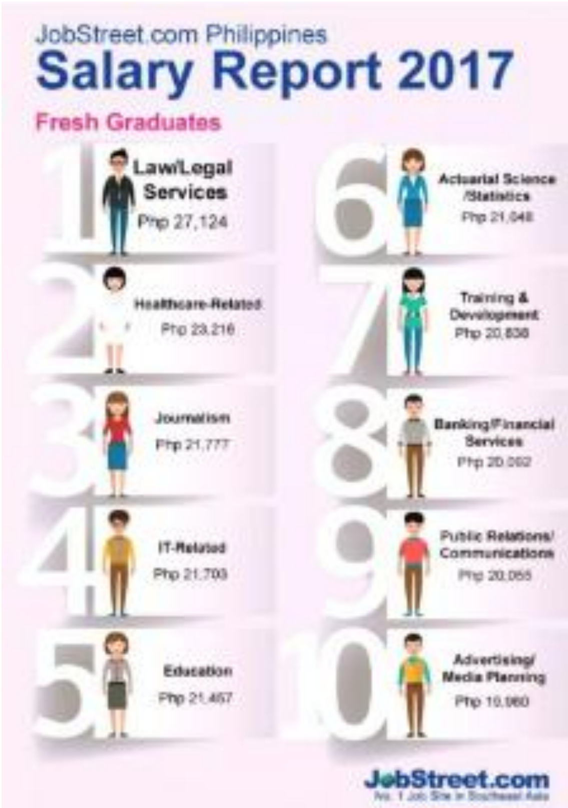
Source - College Salary Report

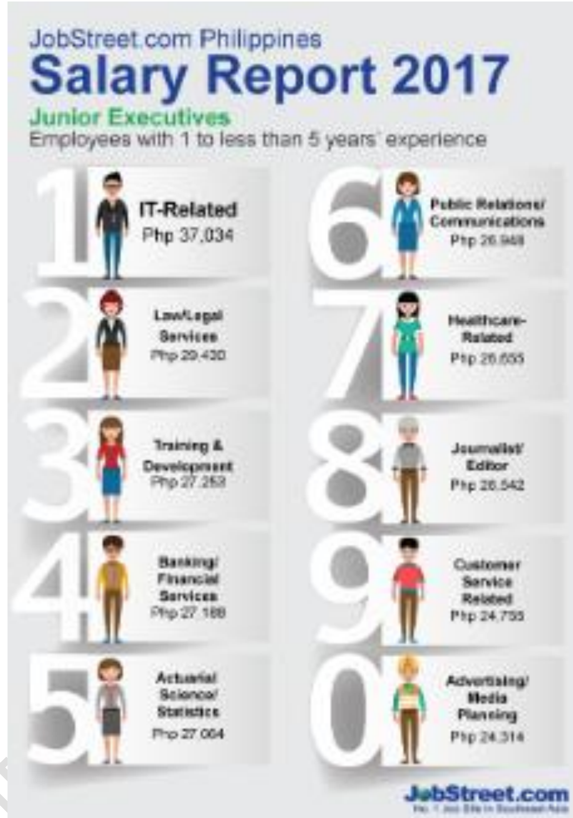
Source: learncomputerscienceonline.com



Source: learncomputerscienceonline.com

Salary Outlook in the Philippines





Computer Engineering Salaries in the Philippines

indeed Find Jobs Company Reviews Find Salaries Employers / Post Job

Computer Engineer Salaries in the Philippines

Salary estimated from 30 employee, user, and past and present job advertisement on Indeed in the past 36 months. Last updated: June 26, 2019

Average salary

Php 22,665 per month

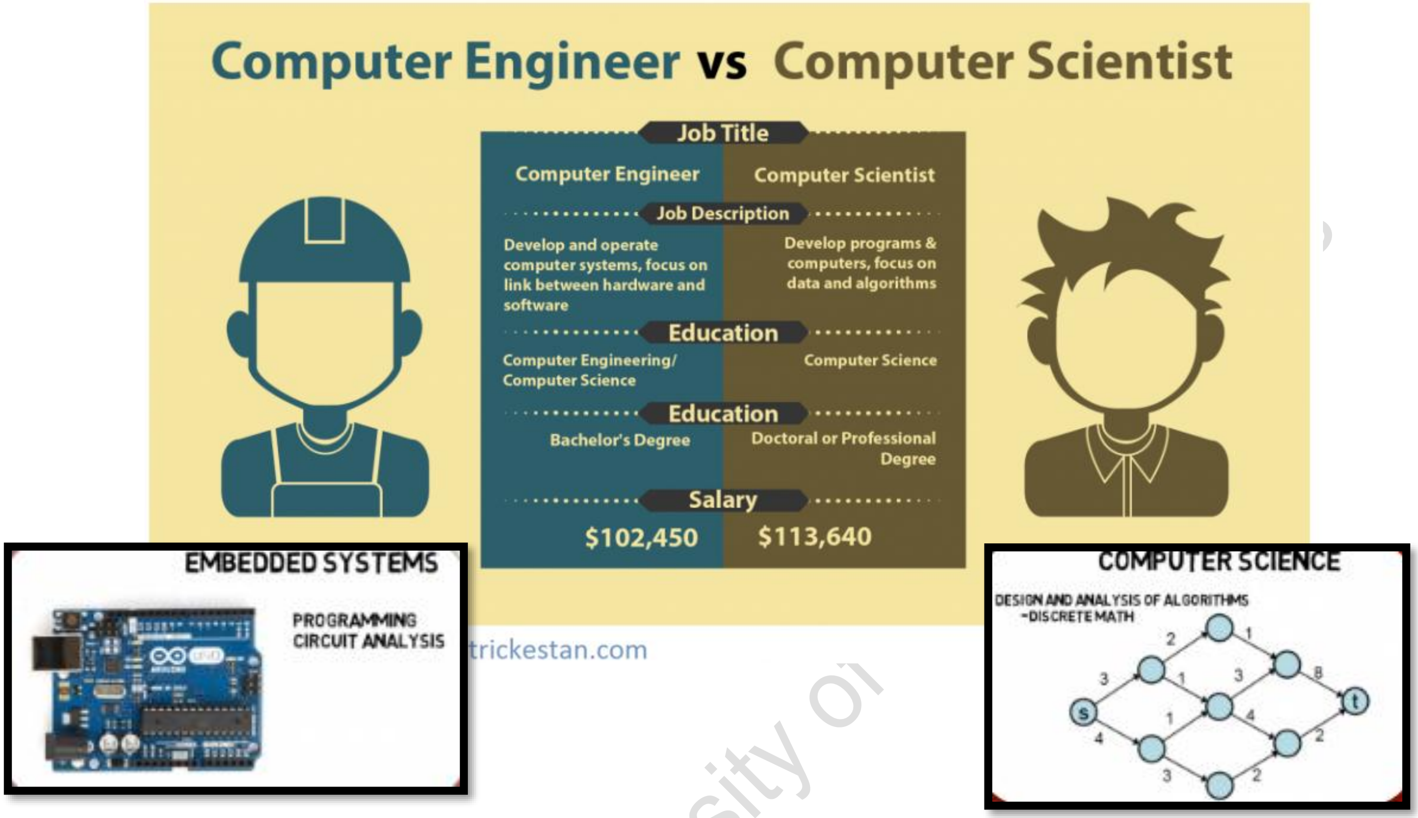
Salary Distribution

Php 8,000 Php 45,000

How much does a Computer Engineer make in the Philippines?

The average salary for a Computer Engineer is **Php 22,665** per month in the Philippines. Salary estimates are based on 30 salaries submitted anonymously to Indeed by Computer Engineer employees, users, and collected from past and present job advertisements on Indeed in the past 36 months.

Computer Engineering and Computer Scientist



What does Computer Engineer Do?

Computer Engineer has two major specialty:

- Computer Software Engineer

  They develop, design and test software or construct, maintain computer networks, programs for companies. They create computer interfaces, operating systems, new programs and applications such as desktops, smartphones, tablets.

- Computer Hardware

  They research, develop and test hardware or computer equipment including chip-set, mainboard… etc. They ensure that their hardware work properly with the latest software technology.

Where Do Computer Engineers Work and In What Roles?

- Computer Engineering Degree holders often land positions in enterprises as software engineers or programmers, software developers, or project managers. Computer engineers benefit from a wide range of industries employing their services, including computer manufactures, financial services, defense contractors, consulting, manufacturing, consumer good, transportation and more.

- Computer engineers are also sometimes referred to as IT engineers, system engineers, computer hardware engineers, or computer software engineers, depending on their industry and area of specialization

Typical Job Duties and Responsibilities

Computer Engineer requires individuals to function effectively within a team and for some projects, with minimal supervision.

Job Opportunities and Position

- Develop computer algorithms and advance systems
- Design and develop scripts and/or tools for integration deployments
- Maintain monitoring systems to ensure high availability
- Create and maintain runbook, technical and operational documents
- Research, develop and prototype advanced hardware and software technologies
- Apply machine learning to computer vision problems
- Conceive proof-of-concept prototypes
- Analyze and improve efficiency, scalability, and stability of various deployed systems
- Automation of provisioning of internal and cloud infrastructure
- Design and develop scripts and tools for integration and deployments
- Work closely with development team in supporting infrastructure requests

Job Opportunities for Graduates of BS in Computer Engineering Graduates

Here are several job opportunities for graduates of BS in Computer Engineering graduates:

1. **Software Engineer/Developer:** Design, develop, and maintain software applications.
2. **Systems Engineer:** Oversee and manage the installation, configuration, and maintenance of hardware and software systems.
3. **Network Engineer:** Design, implement, and manage computer networks.
4. **Embedded Systems Engineer:** Develop software for embedded systems used in devices like cars, appliances, and medical equipment.
5. **Web Developer:** Create and maintain websites, ensuring functionality, user experience, and aesthetics.
6. **Database Administrator:** Manage and maintain databases, ensuring data security, integrity, and availability.

7. **Cybersecurity Analyst:** Protect an organization’s computer systems and networks from cyber threats.
8. **Cloud Solutions Architect:** Design and manage cloud computing strategies and infrastructure.
9. **Data Scientist/Analyst:** Analyze and interpret complex data to help organizations make informed decisions.
10. **IT Consultant:** Advise organizations on how to use technology to meet their business objectives.
11. **Quality Assurance Engineer**: Ensure that software and systems are reliable, functional, and meet standards through testing.
12. **Hardware Engineer**: Design and test computer hardware components, such as circuit boards and processors.
13. **Technical Support Engineer**: Provide assistance and support to customers and organizations regarding technology and software issues.
14. **Project Manager**: Oversee technology projects, coordinating teams and resources to meet deadlines and goals.
15. **Mobile App Developer**: Create applications for mobile devices, focusing on functionality and user experience.
16. **Research and Development Engineer**: Work on new technologies and innovations in computing and engineering.
17. **Machine Learning Engineer**: Design and implement machine learning models and algorithms.
18. **Game Developer**: Design and create video games across various platforms.
19. **DevOps Engineer**: Work on software development and IT operations, focusing on automation and improving deployment efficiency.
20. **System Analyst**: Analyze and improve IT systems, providing solutions to business problems.

These positions can vary by industry, so it’s a good idea for graduates to explore opportunities in fields such as healthcare, finance, education, or technology startups. *Networking and internships can also enhance job prospects.*

List of BSCpE Elective Based on Track of Specialization

The Bachelor of Science in Computer Engineering in Polytechnic University of the Philippines offers five (5) different specializations. The student would take these courses in their junior year (3rd year) that allocates 12 units in total. These courses are pivotal for the students to gain more knowledge and amplify their skills for the industry.

For Computer Networks Engineering:

List of CpE Electives based on Track of Specializations

Computer Networks Engineering

Course Code	Course Title	Term to be offered	Course Credit	Number of Hours	
				Lecture	Laboratory
CPE	Network Administration	3rd Year – First Semester	3	2	3
CPE	Enterprise Networking	3rd Year- Second Semester	3	2	3
CPE	Introduction to Cybersecurity	4th Year- First Semester	3	2	3
CPE	Cloud Computing	4th Year – Second Semester	3	2	3
	TOTALS		12	8	12

For Machine Learning:

Machine Learning

Course Code	Course Title	Term to be offered	Course Credit	Number of Hours	
				Lecture	Laboratory
CPE	Predictive Analytics Modelling, Simulation and Optimization	3rd Year – First Semester	3	2	3
CPE	Pattern Recognition	3rd Year – Second Semester	3	2	3
CPE	Digital Image Processing	4th Year – First Semester	3	2	3
CPE	Neural Networks and Machine Learning	4th Year – Second Semester	3	2	3
	TOTALS		12	8	12

For Big Data:

Big Data

Course Code	Course Title	Term to be offered	Course Credit	Number of Hours	
				Lecture	Laboratory
CPE	Introduction to Big Data	3rd Year – First Semester	3	2	3
CPE	Big Data Analytics	3rd Year – Second Semester	3	2	3
CPE	Database System Implementation	4th Year – First Semester	3	2	3
CPE	Secure Data Management	4th Year – Second Semester	3	2	3
	TOTALS		12	8	12

For System Development

System Development

Course Code	Course Title	Term to be offered	Course Credit	Number of Hours	
				Lecture	Laboratory
CPE	Enterprise Software Systems	3rd Year – First Semester	3	2	3
CPE	Web and Mobile Systems	3rd Year – Second Semester	3	2	3
CPE	Software Process and Product Quality	4th Year – First Semester	3	2	3
CPE	Trends in Software Development Processes	4th Year – Second Semester	3	2	3
	TOTALS		12	8	12

For Mechatronics:

Mechatronics

Course Code	Course Title	Term to be offered	Course Credit	Number of Hours	
				Lecture	Laboratory
CPE	Principle of Automatic Controls	3rd Year – First Semester	3	2	3
CPE	Design of Electromechanical Machines	3rd Year – Second Semester	3	2	3
CPE	Principle of Instrumentation	4th Year – First Semester	3	2	3
CPE	Design of Autonomous Systems	4th Year – Second Semester	3	2	3
	TOTALS		12	8	12

Greatest Computer Engineers

Source: ICASBLOGADMIN

Tim Berners Lee

Sir Timothy Berners Lee is a famous British computer scientist, who created the **World Wide Web or the "internet"** as we know it. In November 1989, Lee and his team successfully implemented the first communication between Hypertext Transfer Protocol (HTTP) client and a server, opening the gateway to sharing information across the virtual world. For his achievement, Lee was awarded the Queen Elizabeth Prize for Engineering and inducted into the World Wide Web Hall of Fame, where he is one of only six members.

Dennis Ritchie

Known as the inventor of the **C programming language** and the **UNIX operating system**, Ritchie was celebrated and oft-awarded computer engineer. With his pioneering work, he paved the way for the creation of various software application, embedded system development, and new-age operating systems like Windows and Linux. Ritchie passed away in 2011, and the Fedora Linus 16 was released in his memory.

Linus Benedict Torvalds

A Finnish-American computer engineer, Torvalds was the dreamer behind the development of the **Linux kernel.** Due to his immensely important contribution to the project, he was later offered the role of the project chief architect and is currently the coordinator for the kernel project. In 2012, Torvalds was awarded the Millenium Technology Prize by the technology Academy Finland, in honour of the open source operating system he helped develop. Torvaids was also the driving force behind popular distribution version control system **Git**, and **Subsurface** – a diving log software.

Donald Knuth

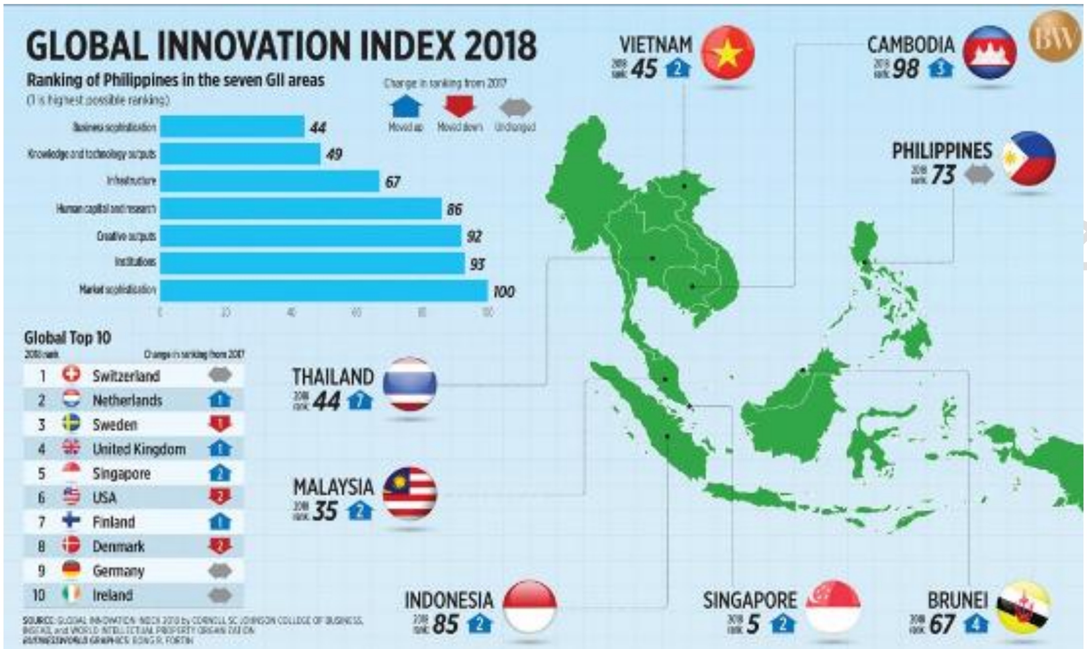
The American scientist Donald Ervin Knuth is dubbed the **"Father of Algorithms".** He retired as a Profressor Emeritus from the prestiguous Stanford University, and is known for his work in developing and systemizing formal mathematical techniques for the analysis of computational complexity of algorithms. His work has been instrumental in advancing the field of theoretical computer science and popularizing the asymptotic notation.

James Gosling

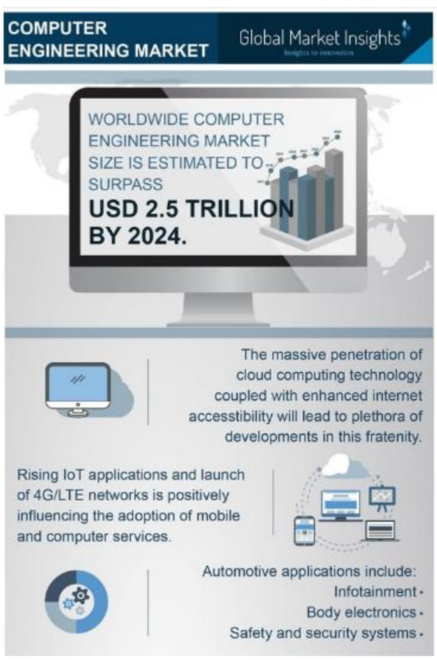
An Officer of the Order of Canada, James Arthur Gosling will forever be remembered as the creator of the **Java Programming Language**. Since 1994, application developers have been using Java and its *'write once, run anywhere'* (WORA) features to code for computer and Android applications. Without Java, the smartphones we use today would probably not be the same! The importance of Gosling's work is reflected in his appointment as the Foreign Associate member of the United States National Academy of Engineering, and by the number of mobile applications flooding the internet today.

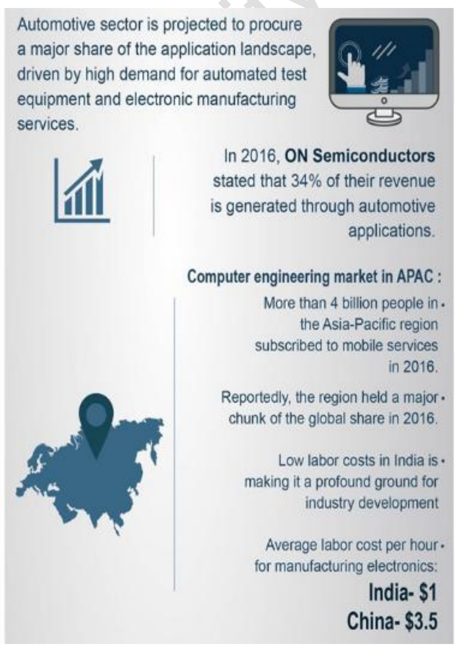
Innovation Trend

Expectations and Opportunities in Philippines



Global Trend Computer Engineering Market





Activity

Aside from the 5 greatest Computer Engineers listed in the previous slide, who are some other people and individuals who work as a computer engineer, computer science, or any IT field, that greatly influence the way we live, and work? Give a 1-page essay.

MODULE 2

Overview of Computer Architecture and Hardware Components

Learning Objectives:

- Understand the fundamental concepts of computer architecture.
- Identify the major hardware components in a computer system.

Content:

1. **Basic Computer Architecture**
    - Von Neumann vs. Harvard Architecture
    - CPU, Memory, Input/Output
2. **Hardware Components Overview**
    - Central Processing Unit (CPU)
    - Memory: RAM, ROM, Cache
    - Storage Devices (HDD, SSD, NVMe)
    - Peripherals and I/O Devices (Keyboard, Mouse, Display)
    - Power Supply, Motherboard

Activities:

- Diagram labeling of major hardware components.
- Hands-on identification of computer parts.

References:

Stallings, W. (2021). Computer Organization and Architecture. Pearson.

Patterson, D. A., & Hennessy, J. L. (2020). Computer Organization and Design RISC-V Edition: The Hardware/Software Interface. Morgan Kaufmann.

Brown, S., & Vranesic, Z. (2018). Fundamentals of Digital Logic with VHDL Design. McGraw-Hill Education.

MODULE 3

Evolution of Computing Systems and Processors

Learning Objectives:

- Explore the historical development of computing systems and processors.
- Understand the trends in microprocessor evolution.

Content:

1. **History of Computing**
   - Early Mechanical and Electrical Computers (ENIAC, UNIVAC)
   - First, Second, and Third Generation Computers

2. **Evolution of Processors**
   - From Single-Core to Multi-Core Processors
   - Moore's Law and Its Impact
   - Notable Processor Families: Intel x86, ARM Architecture

Activities:

- Timeline creation of major processor releases and their capabilities.
- Case study: Comparison between early Intel processors and modern ARM chips.

Evolution Of Computing Systems And Processors

The evolution of computing systems and processors has been a dynamic journey, from early mechanical devices to today's advanced multi-core processors. This evolution is marked by significant milestones in technology, architecture, and performance improvements, all of which have transformed how we compute. Below is a detailed discussion of this evolution:

---

1. **Early Mechanical Computers (Pre-1940s)**

The first computing devices were mechanical systems designed to perform simple arithmetic or store data. These include:

- **Abacus (circa 2500 BC)**: The earliest known computing device, which allowed users to perform basic arithmetic operations by sliding beads along wires.
- **Charles Babbage's Analytical Engine (1837)**: Often considered the precursor to modern computers, Babbage's design for the **Analytical Engine** introduced concepts like the **CPU** and **memory**. Though never fully built, the engine was

capable of conditional branching and loops, resembling modern computational logic.

- **Herman Hollerith's Punch Card System (1890)**: Used for the 1890 U.S. Census, this mechanical system automated the processing of data using punched cards. Hollerith's work laid the foundation for the future of data processing and led to the formation of IBM.

---

2. **The First Generation: Vacuum Tube Computers (1940s-1950s)**

The first electronic computers used **vacuum tubes** as their primary components for computation. These computers were large, power-hungry, and unreliable but marked a significant leap forward in computational power. Notable systems include:

- **ENIAC (Electronic Numerical Integrator and Computer)**: Built in 1945, it is considered the first general-purpose electronic digital computer. ENIAC used approximately 17,000 vacuum tubes and could perform complex calculations much faster than its mechanical predecessors.

- **UNIVAC I (1951)**: Developed by the same team that built ENIAC, UNIVAC I was the first commercially available computer in the United States. It gained fame for correctly predicting the outcome of the 1952 U.S. presidential election.

Processor Characteristics:

- **Vacuum Tubes**: These were large and fragile, leading to frequent failures. They worked by controlling electric current between electrodes in a vacuum.

- **Clock Speeds**: Early vacuum tube computers had relatively slow clock speeds, measured in kilohertz (KHz).

- **Storage**: Used punch cards and magnetic drums for storage.

---

3. **The Second Generation: Transistor Computers (1950s-1960s)**

The invention of the **transistor** at Bell Labs in 1947 by John Bardeen, Walter Brattain, and William Shockley revolutionized computing. By the 1950s, transistors began replacing vacuum tubes in computers.

- **IBM 1401 (1959)**: One of the most popular computers of its era, the IBM 1401 used transistors instead of vacuum tubes, making it smaller, more reliable, and faster.

- **DEC PDP-1 (1960)**: Known for being an interactive system, the PDP-1 introduced concepts such as the **time-sharing system**, which allowed multiple users to use a single computer simultaneously.

Processor Characteristics:

- **Transistors**: Smaller, more reliable, and much more energy-efficient than vacuum tubes, transistors significantly reduced the size of computers.
- **Performance Improvement**: Transistor-based processors operated at megahertz (MHz) speeds, far surpassing vacuum tube systems.
- **Storage**: Used magnetic core memory, which allowed for faster access and higher capacity.

---

4. **The Third Generation: Integrated Circuit (IC) Computers (1960s-1970s)**

The development of **Integrated Circuits (ICs)**, or microchips, during the 1960s allowed for the integration of thousands of transistors on a single silicon chip. This innovation drastically reduced the size and cost of computers while improving performance.

- **IBM System/360 (1964)**: One of the first computers to use IC technology, the System/360 was notable for its ability to run various software programs on the same machine, a concept known as **backward compatibility**.
- **Intel 4004 (1971)**: The first commercially available microprocessor, the Intel 4004, packed 2,300 transistors on a single chip and could perform 60,000 operations per second.

Processor Characteristics:

- **Integrated Circuits**: The use of ICs allowed for far more transistors to be packed onto a single chip, drastically increasing computational power and reducing costs.
- **Microprocessors**: The introduction of microprocessors like the Intel 4004 marked the beginning of modern processor design.
- **Speed**: Processors in this era operated at megahertz speeds, with much greater reliability than previous generations.
- **Storage**: Introduction of **magnetic disks** as a primary storage device, improving capacity and access times.

5. **The Fourth Generation: Microprocessor-Based Systems (1970s-1980s)**

The 1970s saw the rise of microprocessors—single-chip CPUs that became the foundation for personal computers (PCs). The microprocessor revolutionized computing, making it accessible to individuals and small businesses.

- **Intel 8080 (1974)**: The Intel 8080 was one of the first commercially successful microprocessors. It powered early personal computers like the **Altair 8800**, a precursor to modern PCs.
- **Apple II (1977)**: Apple's introduction of the Apple II marked a key moment in the personal computing revolution. Powered by a 1 MHz MOS Technology 6502 processor, the Apple II was one of the first mass-produced personal computers.
- **IBM PC (1981)**: Powered by the Intel 8088 processor, the IBM PC set a standard for personal computing and led to the widespread adoption of computers in homes and offices.

Processor Characteristics:

- **Microprocessors**: Processors were now integrated onto a single chip. Early processors had limited speed and could only handle simple instructions, but their architecture set the stage for future advancements.
- **Clock Speeds**: Processors reached speeds of up to several megahertz (MHz).
- **Storage**: The **floppy disk** was introduced, allowing for portable data storage and transfer.

---

6. **The Fifth Generation: Rise of Multi-Core and Parallel Processing (1990s-Present)**

The late 20th and early 21st centuries saw exponential growth in processor speed and capabilities, leading to innovations like **multi-core processors**, **parallel processing**, and **quantum computing**.

- **Intel Pentium Series (1993)**: The Pentium series was a milestone in the evolution of consumer processors, with clock speeds reaching hundreds of MHz. It introduced the **superscalar architecture**, allowing multiple instructions to be processed simultaneously.
- **AMD Athlon and Intel Core Series (2000s)**: These processors brought **multi-core** technology to the mainstream. Instead of increasing clock speeds, processors began adding more cores, allowing for parallel processing and significant improvements in multitasking.

- **Apple M1 Chip (2020)**: Apple's introduction of the **M1** chip marked a major shift in processor architecture. Built on the **ARM** architecture, it integrated the CPU, GPU, and memory into a single system-on-a-chip (SoC), significantly improving energy efficiency and performance.

Processor Characteristics:

- **Multi-Core Architecture**: Modern processors feature multiple cores on a single chip, allowing for parallel processing and more efficient multitasking.
- **Increased Clock Speeds**: While clock speed improvements have slowed due to physical limitations (e.g., heat), processors now operate at gigahertz (GHz) speeds.
- **Quantum Computing**: Though still in its infancy, quantum computing represents the next frontier, promising to solve complex problems that are currently beyond the reach of classical computers.

Summary

The evolution of computing systems and processors has been characterized by advancements in technology that have transformed the scale, speed, and accessibility of computing. From mechanical devices to the first general-purpose computers using vacuum tubes, the transition to transistors, integrated circuits, and microprocessors has led to exponential increases in performance. The rise of multi-core processors, parallel processing, and the potential of quantum computing continues to shape the future of technology.

Each leap in processor technology has made computers faster, smaller, and more affordable, leading to the ubiquity of computing in everyday life today.

References

Smotherman, M. (2023). *History of Computing and Processors*. Computing History Journal.

Flynn, M., & Hung, E. (2022). *Computer System Design: System-on-Chip Design and Evolution*. Springer.

Hennessy, J. L., & Patterson, D. A. (2022). *Computer Architecture: A Quantitative Approach* (6th ed.). Morgan Kaufmann.

- *This comprehensive textbook provides an in-depth analysis of computer architecture and the evolution of processor design, emphasizing quantitative performance evaluation.*

Patterson, D. A., & Hennessy, J. L. (2020). *Computer Organization and Design RISC-V Edition: The Hardware Software Interface*. Morgan Kaufmann.

- *Focuses on modern processor design using the RISC-V architecture, offering insights into the advancements in microprocessor technology over recent years.*

Sze, V., Chen, Y.-H., Yang, T. J., & Emer, J. S. (2020). *Efficient Processing of Deep Neural Networks: A Tutorial and Survey*. Proceedings of the IEEE, 103(4), 679–702.

- *Explores the impact of deep learning on processor design and the subsequent evolution of computing systems to accommodate AI workloads.*

Heath, T., & Forsythe, P. (2019). *The Evolution of Microprocessors*. IEEE Annals of the History of Computing, 41(2), 24-35.

- *Provides a historical perspective on microprocessor development, highlighting key milestones and technological breakthroughs from the late 20th century to the present.*

Apple Inc. (2021). *Apple M1 Chip Architecture Overview*. Apple Technical Documentation.

- *Details the architecture of Apple's M1 chip, illustrating the shift towards system-on-a-chip (SoC) designs and their implications for computing performance and efficiency.*

AMD Inc. (2022). *Zen Architecture Overview*. AMD Whitepapers.

- *Discusses the advancements in AMD's Zen architecture, including multi-core processing and energy-efficient design strategies that have influenced modern CPU development.*

Intel Corporation. (2023). *Intel Core Processor Architectures*. Intel Developer Zone.

- *Outlines the latest developments in Intel's Core processor line, focusing on innovations in multi-core design, parallel processing, and integration with other system components.*

Marr, B. (2021). *The Evolution of Quantum Computing: Past, Present, and Future*. Nature Electronics, 4(2), 68-72.

- *Examines the progression of quantum computing technologies and their potential to revolutionize traditional processor architectures and computing paradigms.*

Vahid, F., & Givargis, T. (2019). *Embedded Systems Design: A Unified Hardware/Software Introduction*. Wiley.

- *While focused on embedded systems, this book provides valuable insights into the integration of microcontrollers and processors within specialized computing environments.*

Sze, V., & Chen, Y. (2021). *Artificial Intelligence and the Future of Computing Systems*. IEEE Transactions on Computers, 70(1), 10-15.

- *Discusses how artificial intelligence is driving the evolution of computing systems and processors, highlighting trends such as specialized AI accelerators and neuromorphic computing.*

MODULE 4

Introduction to Embedded Systems

Learning Objectives:

By the end of the lesson, students will be able to:

1. Define what an embedded system is.
2. Understand the basic components of an embedded system.
3. Identify real-world applications of embedded systems.
4. Differentiate between general-purpose and embedded systems.
5. Explore career opportunities related to embedded systems.

Content:

1. **Basic Computer Architecture**
   - Von Neumann vs. Harvard Architecture
   - CPU, Memory, Input/Output
2. **Hardware Components Overview**
   - Central Processing Unit (CPU)
   - Memory: RAM, ROM, Cache
   - Storage Devices (HDD, SSD, NVMe)
   - Peripherals and I/O Devices (Keyboard, Mouse, Display)
   - Power Supply, Motherboard

Activities:

- Group discussion on the role of embedded systems in everyday life.
- Project: Building a simple embedded system using Arduino or Raspberry Pi.

Introduction to Embedded Systems

An embedded system is a specialized computing system designed to perform dedicated functions, often within larger systems. Unlike general-purpose computers, which can run a variety of software programs, embedded systems are optimized to perform specific tasks, typically with real-time constraints. They are integrated into a wide range of devices, from consumer electronics like smartphones to industrial machines and medical equipment.

1. Definition and Characteristics

An **embedded system** is a combination of hardware and software designed to perform a specific function or set of functions. The system typically operates within a broader mechanical or electrical system and interacts with the external environment through sensors and actuators.

Key Characteristics:

- **Task-Specific**: Embedded systems are designed for a specific task. For example, an embedded system in a washing machine only controls the washing process.
- **Real-Time Operation**: Many embedded systems are real-time systems, meaning they must process data and respond within a strict time frame. This is crucial in applications like automotive systems, where delays could lead to failures.
- **Resource Constraints**: Embedded systems often operate with limited resources, such as memory, processing power, and energy. Optimizing the system's performance within these constraints is crucial.
- **Low Power Consumption**: Many embedded systems are designed to be energy-efficient, especially in battery-powered devices like smartphones or medical implants.
- **Reliability and Stability**: Because embedded systems often operate in critical environments (like medical devices or aerospace systems), they must be highly reliable and stable over long periods.

---

2. Components of an Embedded System

An embedded system typically comprises several key components:

2.1. Hardware Components

- **Microcontroller (MCU) or Microprocessor (MPU)**: The central processing unit (CPU) in an embedded system is usually a microcontroller or microprocessor. A microcontroller, like the **ARM Cortex-M** series, integrates a CPU, memory, and input/output (I/O) interfaces on a single chip. Microprocessors like the **Intel Atom** may also be used in more complex embedded systems.
- **Memory**: Embedded systems typically have a combination of **read-only memory (ROM)** for storing the operating software and **random-access memory (RAM)** for temporary data storage.

- **Input/Output Interfaces**: Embedded systems interact with the external world via sensors (inputs) and actuators or displays (outputs). Examples include a temperature sensor in a smart thermostat or a display in a digital camera.
- **Timers and Counters**: Embedded systems often include built-in timers to keep track of operations, manage delays, and handle real-time tasks.
- **Power Supply**: Embedded systems often require energy-efficient power sources, such as batteries or renewable energy in IoT (Internet of Things) devices.

2.2. Software Components

- **Firmware**: Firmware is the specialized software written for the embedded system, stored in non-volatile memory (like flash memory). It defines how the system operates and interacts with its hardware components.
- **Real-Time Operating System (RTOS)**: Many embedded systems use an RTOS to manage tasks, prioritize processes, and handle real-time constraints. Examples include **FreeRTOS** and **VxWorks**.
- **Device Drivers**: Software that allows the embedded system to communicate with hardware peripherals like sensors, actuators, and communication modules.

---

3. Types of Embedded Systems

Embedded systems can be classified based on various factors, such as performance, complexity, and application. The most common types include:

3.1. Based on Performance and Functional Requirements

- **Real-Time Embedded Systems**: These systems must respond to inputs or events within a specified time frame. They can be further divided into:
  - **Hard Real-Time Systems**: Require a strict time-bound response, as in pacemakers or automotive control systems.
  - **Soft Real-Time Systems**: Allow for some flexibility in timing. For example, streaming media players tolerate occasional delays.
- **Standalone Embedded Systems**: These systems can function independently without external control. Examples include digital cameras and washing machines.
- **Networked Embedded Systems**: These systems are connected to a network, often as part of the **Internet of Things (IoT)**. Examples include smart thermostats and home automation systems.

- **Mobile Embedded Systems**: Embedded systems designed for portability, such as smartphones, wearable devices, and tablets. They are optimized for low power consumption.

3.2 Based on Complexity

- **Small-Scale Embedded Systems**: These systems use 8-bit or 16-bit microcontrollers, have simple functionalities, and are relatively inexpensive. An example is an embedded system in a home appliance.
- **Medium-Scale Embedded Systems**: These systems use 16-bit or 32-bit microcontrollers and are more complex. They are found in industrial automation or automotive applications.
- **Large-Scale Embedded Systems**: These systems use high-performance processors, multiple interfaces, and an RTOS. They are found in applications like air traffic control or autonomous vehicles.

---

4. Applications of Embedded Systems

Embedded systems are present in almost every sector of modern technology. Some notable applications include:

4.1. Consumer Electronics

- **Smartphones**: Embedded systems manage tasks like processing user inputs, controlling cameras, and handling communications.
- **Smart Home Devices**: Thermostats, security systems, and smart lighting all use embedded systems for automation and remote control.

4.2. Automotive

- **Anti-lock Braking Systems (ABS)**: Embedded systems control braking pressure to prevent wheels from locking during emergency stops.
- **Infotainment Systems**: Modern vehicles include embedded systems to manage entertainment, navigation, and communication functions.

4.3. Healthcare

- **Medical Devices**: Embedded systems are used in devices like pacemakers, insulin pumps, and MRI machines, where real-time processing is critical.

4.4. Industrial Automation

- **Robotics**: Embedded systems control the movement and actions of industrial robots in assembly lines.

- **Process Control**: Systems that monitor and control industrial processes (e.g., temperature regulation) are embedded in many factories.

4.5. Aerospace

- **Flight Control Systems**: Aircraft rely on embedded systems to manage navigation, engine control, and safety systems.

---

5. Design Challenges in Embedded Systems

Embedded systems often come with unique design challenges, especially compared to general-purpose computers:

- **Power Efficiency**: Many embedded systems, especially those used in mobile and IoT devices, must operate on limited power sources such as batteries, requiring energy-efficient designs.
- **Real-Time Constraints**: Meeting strict timing requirements, especially in real-time systems, is crucial. Delays or failures in timing can result in malfunctioning or catastrophic results (e.g., in medical devices or automotive systems).
- **Security**: As embedded systems become increasingly connected, security concerns become critical, particularly in IoT devices that are susceptible to hacking and data breaches.
- **Scalability and Cost**: Designers must balance performance with cost, ensuring that the embedded system is affordable for its intended market while meeting the necessary functional requirements.

---

6. Trends in Embedded Systems

Several trends are shaping the future of embedded systems:

- **Internet of Things (IoT)**: The rise of IoT has led to an explosion in networked embedded systems that collect, transmit, and process data. Smart homes, smart cities, and industrial IoT are key areas of growth.
- **Edge Computing**: Instead of sending all data to centralized cloud servers for processing, edge computing allows embedded systems to process data locally, reducing latency and bandwidth usage.
- **AI and Machine Learning**: Embedded systems are increasingly incorporating artificial intelligence (AI) to perform tasks like image recognition and natural

language processing. This is evident in smart devices like voice assistants (e.g., Amazon Alexa).

- **Wearable Technology**: Embedded systems are playing a pivotal role in the development of health and fitness tracking devices, augmented reality glasses, and other wearable tech.
- **Energy Harvesting**: For IoT devices, energy harvesting from sources like solar power or kinetic energy is an emerging trend to power embedded systems sustainably.

Conclusion

Embedded systems are the backbone of modern technology, driving innovation across industries. Their task-specific nature, real-time operation, and resource-constrained environments present unique design challenges, but also immense opportunities. As technology continues to advance, embedded systems will become even more integral to fields such as IoT, artificial intelligence, and edge computing.

References

Kamal, R. (2021). *Embedded Systems: Architecture, Programming and Design*. McGraw-Hill Education.

Wolf, W. (2022). *Computers as Components: Principles of Embedded Computing System Design* (3rd ed.). Morgan Kaufmann.

- *This textbook offers a comprehensive introduction to embedded systems, covering both hardware and software aspects. It emphasizes the integration of computing components within larger systems and provides practical design examples.*

Barr, M., & Massa, A. (2021). *Programming Embedded Systems: With C and GNU Development Tools* (3rd ed.). O'Reilly Media.

- *A practical guide to programming embedded systems using the C language and GNU tools. The book includes numerous examples and exercises that help readers understand the fundamentals of embedded software development.*

Lyons, R. (2020). *Understanding Embedded Systems* (4th ed.). Elsevier.

- *This book provides a clear and thorough introduction to embedded systems, focusing on both the hardware and software components. It covers real-time operating systems, interfacing, and system design principles.*

Vahid, F., & Givargis, T. (2019). *Embedded Systems: An Integrated Approach* (3rd ed.). Wiley.

- *An integrated approach to embedded systems design, this book covers the essentials of both hardware and software. It includes case studies and practical examples that illustrate key concepts in real-world applications.*

Mahmood, A. (2021). *Introduction to Embedded Systems: Using ANSI C and the Arduino Development Environment*. CRC Press.

- *This introductory text focuses on embedded systems programming using ANSI C and the Arduino platform. It is ideal for beginners and includes hands-on projects that reinforce learning through practical application*

MODULE 5

Overview of Integrated Circuits and Microcontrollers

Learning Objectives:

- Learn the fundamentals of integrated circuits (ICs) and microcontrollers (MCUs).
- Understand the role of microcontrollers in embedded systems.

Content:

1. **Introduction to Integrated Circuits**
    - Overview of IC Manufacturing
    - Types of ICs: Analog, Digital, and Mixed-Signal
    - Moore's Law and VLSI (Very-Large-Scale Integration)

2. **Introduction to Microcontrollers**
    - Microcontroller Architecture
    - Applications of Microcontrollers in Embedded Systems
    - Popular Microcontroller Families: AVR, ARM Cortex, PIC

Activities:

- Hands-on lab: Introduction to programming a microcontroller (e.g., Arduino).
- Group research project: Investigating the evolution of ICs and microcontrollers.

Overview of Integrated Circuits and Microcontrollers

Integrated Circuits (ICs) are essential components of modern electronics. An integrated circuit is a semiconductor device that contains multiple electronic components, such as transistors, diodes, resistors, and capacitors, all miniaturized and interconnected on a single chip. These components work together to perform a specific function, making ICs the backbone of virtually all modern electronic devices, from computers to smartphones.

ICs can be classified into various types based on their functions, such as:

- **Analog ICs**: Process continuous signals (e.g., amplifiers, voltage regulators).
- **Digital ICs**: Handle digital signals (e.g., microprocessors, memory chips).
- **Mixed-Signal ICs**: Handle both analog and digital signals (e.g., analog-to-digital converters).

Microcontrollers (MCUs) are a specific type of digital integrated circuit that integrates a processor, memory, and input/output (I/O) peripherals on a single chip. Microcontrollers are designed to perform specific control functions in embedded systems. They are widely used in various applications, including automotive systems, industrial automation, consumer electronics, and home appliances.

Key features of microcontrollers include:

- **Processor (CPU)**: The central processing unit that executes instructions.
- **Memory**: Includes both volatile memory (RAM) for temporary data storage and non-volatile memory (flash/EEPROM) for storing the program code.
- **I/O Ports**: Allow the microcontroller to interact with external devices such as sensors, motors, and displays.
- **Timers/Counters**: Essential for handling real-time tasks.
- **Communication Interfaces**: Support protocols like UART, SPI, and I2C for data exchange with other devices.

Microcontrollers are often used in applications where cost, power consumption, and size are critical. Popular microcontroller families include the **ARM Cortex-M series**, **Atmel AVR (used in Arduino)**, and **PIC microcontrollers** from Microchip Technology.

References

Sedra, A. S., & Smith, K. C. (2020). *Microelectronic Circuits* (8th ed.). Oxford University Press.

- *A widely used textbook that provides an in-depth analysis of integrated circuits, including analog and digital components, with detailed explanations of their design and applications.*

Mazidi, M. A., McKinlay, R. D., & Causey, D. (2022). *PIC Microcontroller and Embedded Systems: Using Assembly and C for PIC18* (3rd ed.). Pearson.

- *A detailed guide to microcontrollers, focusing on the PIC18 series. It covers both hardware and software aspects, making it a comprehensive resource for embedded system designers.*

Peatman, J. B. (2019). *Embedded Design with the PIC18F452 Microcontroller*. Prentice Hall.

- *Offers a detailed look into microcontroller design, including practical applications of embedded systems using the PIC18F452 microcontroller.*

Hennessy, J. L., & Patterson, D. A. (2021). *Computer Organization and Design: The Hardware/Software Interface* (5th ed.). Morgan Kaufmann.

- *This textbook provides a solid foundation in computer architecture, including the role of integrated circuits and microcontrollers in system design.*

Weste, N. H. E., & Harris, D. M. (2020). *CMOS VLSI Design: A Circuits and Systems Perspective* (4th ed.). Pearson.

- *An authoritative book on the design and operation of integrated circuits using CMOS technology, a key component in modern microcontroller design.*

MODULE 6-7

Software Engineering in Computer Engineering

Learning Objectives:

By the end of this lesson, students will be able to:

- Understand the fundamentals of software development in Computer Engineering.
- Recognize the role of programming in computer systems design.
- Explain the functions of operating systems.
- Differentiate between software and hardware in system design.

Software Engineering in Computer Engineering

I. **Introduction to Software Development in Computer Engineering**

1. **Definition of Software Development**:
    - Software development involves creating, designing, testing, and maintaining software programs or applications.
    - In **Computer Engineering**, software development is crucial for building systems, applications, and embedded systems.
2. **Key Phases of Software Development**:
    - **Requirement Analysis**: Understanding what the software must do.
    - **Design**: Structuring how the software will work.
    - **Implementation (Coding)**: Writing the actual code.
    - **Testing**: Ensuring the software functions as expected.
    - **Deployment**: Making the software available for use.
    - **Maintenance**: Updating and fixing the software over time.
3. **Tools Used in Software Development**:
    - **IDEs (Integrated Development Environments)**: Tools like Visual Studio, Eclipse.
    - **Version Control Systems**: Git for tracking changes in code.
    - **Programming Languages**: C, C++, Python, Java.

II. **Role of Programming in Computer Systems Design**

1. **Programming as the Backbone of Computer Systems**:
    - Programming is essential for **defining the behavior** of a system and automating tasks.

- **Embedded Systems**: Microcontrollers rely on software programming to control operations in devices (e.g., sensors, actuators).

2. **Levels of Programming**:
    - **Low-Level Programming**: Close to hardware (e.g., Assembly, C).
    - **High-Level Programming**: More abstract, dealing with logic and operations (e.g., Python, Java).
3. **Application of Programming in System Design**:
    - **Operating Systems Development**: Operating systems like Linux and Windows are built using low-level programming.
    - **System Performance**: Efficient code can lead to faster and more reliable systems.

III. **Overview of Operating Systems and Their Functions**

1. **Definition of Operating Systems (OS)**:
    - An **Operating System** is system software that manages hardware and software resources and provides common services for computer programs.
2. **Key Functions of Operating Systems**:
    - **Process Management**: Manages the execution of processes.
    - **Memory Management**: Allocates and deallocates memory to programs.
    - **File System Management**: Organizes and stores files on storage devices.
    - **Device Management**: Controls peripherals like printers, keyboards, and monitors.
    - **Security and Access Control**: Ensures data privacy and restricts unauthorized access.
3. **Types of Operating Systems**:
    - **Real-Time OS (RTOS)**: Used in embedded systems where timely execution is critical (e.g., automotive systems).
    - **General-Purpose OS**: For personal computers (e.g., Windows, macOS, Linux).

IV. **Software vs. Hardware in System Design**

1. **Definition of Hardware**:
    - **Hardware** refers to the physical components of a computer system, such as the processor (CPU), memory (RAM), storage (HDD/SSD), and peripherals.

2. **Definition of Software**:
   - **Software** is a set of instructions or code that tells the hardware how to perform tasks. It includes operating systems, applications, and embedded programs.
3. **Comparison:**
   - **Hardware** is tangible, while **software** is intangible.
   - **Hardware** performs mechanical and electronic tasks, while **software** executes logical operations.
   - **Software** can be updated easily without changing the physical system, whereas **hardware** may need to be replaced for upgrades.
4. **Integration in System Design**:
   - Computer systems require both **hardware and software** to function effectively.
   - **Hardware without software** is non-functional, while **software without hardware** has no medium to execute its instructions.
   - **Embedded Systems**: Often, software is embedded directly into the hardware (firmware), making them highly interdependent.

V. Summary and Q&A

- **Recap Key Points**:
  - Introduction to software development and its importance in Computer Engineering.
  - Role of programming in designing and controlling computer systems.
  - Operating systems and their critical functions.
  - The distinction between hardware and software in system design.
- **Questions**: Encourage students to ask questions and provide clarifications.

References

Stallings, W. (2020). *Operating Systems: Internals and Design Principles* (9th ed.). Pearson.

Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.

Tanenbaum, A. S., & Bos, H. (2021). *Modern Operating Systems* (4th ed.). Pearson.

Sommerville, I. (2020). *Software Engineering* (11th ed.). Pearson.

Hennessy, J. L., & Patterson, D. A. (2019). *Computer Architecture: A Quantitative Approach* (6th ed.). Morgan Kaufmann.

MODULE 8-9

Networking and Communication Systems

Learning Objectives:

By the end of this lesson, students will be able to:

- Understand the fundamentals of computer networks and the Internet.
- Learn the basics of data communication and different network topologies.
- Recognize the role of networking in modern computer engineering.
- Differentiate between wired and wireless networks and identify emerging networking technologies.

Networking and Communication Systems

I. Introduction to Computer Networks and the Internet (15 minutes)

1. **Definition of Computer Networks**:
    - A **computer network** is a group of interconnected devices that share resources and data using communication protocols.
    - Networks range from **local area networks (LANs)**, which cover small geographical areas, to **wide area networks (WANs)**, such as the Internet, which span the globe.
2. **The Internet**:
    - The **Internet** is the world's largest public WAN, connecting billions of devices worldwide using the **TCP/IP protocol**.
    - It enables the use of services such as **email**, **web browsing**, **cloud storage**, and more.
3. **Key Components**:
    - **Routers**: Forward data between networks.
    - **Switches**: Connect devices within a network.
    - **Servers**: Provide resources to other devices.

II. Basics of Data Communication and Network Topologies (15 minutes)

1. **Data Communication**:
    - Data communication refers to the transfer of data between devices via a communication medium (e.g., copper wires, fiber optics, wireless signals).
    - **Key Elements**:
        - **Sender**: Device that transmits data.
        - **Receiver**: Device that receives the data.

- **Transmission Medium**: Path through which data travels (e.g., cables, air).
- **Protocol**: Set of rules for data transmission (e.g., TCP/IP, HTTP).

III. Role of Networking in Modern Computer Engineering (10 minutes)

1. **Enabling Communication**:
    - Networking is crucial for **communication** between devices, enabling systems to exchange data and share resources such as printers, servers, and storage devices.
2. **Distributed Systems**:
    - **Distributed computing** allows multiple computers to work together, splitting complex tasks across a network (e.g., cloud computing).
3. **Network Security**:
    - Networking plays a critical role in computer security, with protocols like **firewalls** and **VPNs** ensuring safe data transmission.
4. **Internet of Things (IoT)**:
    - Networking enables the IoT, where interconnected smart devices communicate with each other, providing innovations in healthcare, transportation, and industrial automation.

IV. Wired vs. Wireless Networks and Emerging Technologies (15 minutes)

1. **Wired Networks**:
    - Wired networks use physical cables (e.g., Ethernet) to connect devices.
    - **Advantages**: High reliability, faster speeds, and security.
    - **Disadvantages**: Limited mobility and higher installation costs.
2. **Wireless Networks**:
    - Wireless networks use radio waves to transmit data, enabling mobility.
    - **Wi-Fi**, **Bluetooth**, and **cellular networks** (e.g., 4G, 5G) are common wireless technologies.
    - **Advantages**: Flexibility, ease of installation, and mobility.
    - **Disadvantages**: Susceptible to interference, lower speeds compared to wired networks.
3. **Emerging Networking Technologies**:
    - **5G Networks**: High-speed mobile networks offering faster data transmission and low latency.
    - **Software-Defined Networking (SDN)**: Provides centralized control over the network through software, improving flexibility and management.

- **Network Function Virtualization (NFV)**: Allows network functions (e.g., firewalls, routers) to be virtualized, reducing the need for dedicated hardware.

V. Summary and Q&A (5 minutes)

- **Recap Key Points**:
  - The basics of computer networks and the Internet.
  - Key components of data communication and common network topologies.
  - The role of networking in modern computer engineering, from communication to IoT.
  - A comparison of wired vs. wireless networks, and emerging networking technologies like 5G and SDN.
- **Questions**: Open the floor for student questions.

References

Tanenbaum, A. S., & Wetherall, D. J. (2020). *Computer Networks* (6th ed.). Pearson.

Forouzan, B. A. (2021). *Data Communications and Networking* (6th ed.). McGraw-Hill.

Kurose, J. F., & Ross, K. W. (2021). *Computer Networking: A Top-Down Approach* (8th ed.). Pearson.

Olsson, M. (2021). *5G Networks: Concepts, Practices, and Practical Aspects*. Wiley.

MODULE 10-11

Emerging Trends in Computer Engineering

Learning Objectives:

By the end of this lesson, students will be able to:

- Understand how Artificial Intelligence (AI) and Machine Learning (ML) are applied in Computer Engineering.
- Recognize the role of big data and cloud computing in modern systems.
- Explain the impact of the Internet of Things (IoT) on society.
- Explore the significance of robotics and automation in engineering.

Emerging Trends in Computer Engineering

I. **Artificial Intelligence (AI) and Machine Learning (ML) Applications in Computer Engineering**

1. **Introduction to AI and ML**:
   - **Artificial Intelligence (AI)** refers to the simulation of human intelligence in machines that are programmed to think and learn. It allows systems to perform tasks such as **speech recognition**, **decision-making**, and **visual perception**.
   - **Machine Learning (ML)** is a subset of AI, focusing on enabling machines to learn from data without being explicitly programmed. ML algorithms improve automatically as they are exposed to more data.
2. **Applications of AI and ML in Computer Engineering**:
   - **Autonomous Vehicles**: AI is critical in self-driving cars, allowing them to perceive the environment and make driving decisions.
   - **Speech and Image Recognition**: ML models, such as **neural networks**, are used to identify objects in images and convert speech to text.
   - **Predictive Maintenance**: AI systems are used in industrial automation to predict equipment failures and reduce downtime.
   - **AI in Healthcare**: AI models assist in diagnosing diseases and personalizing patient care.
   - **Russell, S., & Norvig, P. (2021).** *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

II. Role of Big Data and Cloud Computing

1. **Introduction to Big Data**:
   - **Big Data** refers to the large volumes of structured and unstructured data generated by various sources, such as social media, sensors, and transaction logs.
   - **Characteristics of Big Data**:
     - **Volume**: Large amount of data.
     - **Velocity**: Speed at which data is generated and processed.
     - **Variety**: Different types of data (e.g., text, video, images).
2. **Cloud Computing**:
   - **Cloud computing** provides on-demand computing services (storage, processing power, and networking) over the internet, enabling flexible resource allocation without needing physical infrastructure.
   - **Types of Cloud Services**:
     - **IaaS (Infrastructure as a Service)**: Offers virtualized computing resources (e.g., AWS EC2).
     - **PaaS (Platform as a Service)**: Provides platforms for developing and deploying applications (e.g., Google App Engine).
     - **SaaS (Software as a Service)**: Delivers software applications over the internet (e.g., Gmail, Dropbox).
3. **Applications in Computer Engineering**:
   - **Data Analytics**: Big data analytics helps in extracting useful insights from large datasets, driving innovation in sectors like healthcare, finance, and marketing.
   - **Cloud-based Development**: Engineers use cloud platforms for software development, collaborative coding, and scalable deployment of applications.

III. IoT (Internet of Things) and Its Impact on Society (30 minutes)

1. **Introduction to IoT**:
   - **The Internet of Things (IoT)** refers to the network of physical devices, vehicles, home appliances, and other items embedded with sensors, software, and connectivity that allows them to collect and exchange data.
   - IoT allows everyday objects to be connected to the internet and interact with the world in intelligent ways.

2. **Impact of IoT on Society**:
    - **Smart Homes**: Devices like thermostats, security cameras, and refrigerators are connected and controlled remotely via smartphones.
    - **Smart Cities**: IoT enables cities to monitor traffic, manage waste, and optimize energy use through connected sensors.
    - **Healthcare**: Wearable devices monitor vital signs, improving patient care by providing real-time data to healthcare professionals.
3. **Security and Privacy Concerns**:
    - With more devices connected to the internet, security vulnerabilities and data privacy issues have increased, requiring robust solutions to protect sensitive data.

IV. Robotics and Automation

1. **Introduction to Robotics**:
    - **Robotics** involves the design, construction, and operation of robots, which are programmable machines capable of performing tasks autonomously or semi-autonomously.
    - Robots are used in various fields, including **manufacturing**, **healthcare**, **exploration**, and **domestic services**.
2. **Automation in Computer Engineering**:
    - **Automation** refers to using control systems, such as computers or robots, to handle tasks with minimal human intervention.
    - Examples of automation include **assembly lines** in factories, **automated testing** in software development, and **data entry automation**.
3. **Impact on the Workforce**:
    - While robotics and automation increase efficiency and reduce costs, they also raise concerns about job displacement in industries that rely heavily on manual labor.
4. **Emerging Trends**:
    - **Collaborative Robots (Cobots)**: Robots designed to work alongside humans in factories.
    - **Autonomous Drones**: Used for surveillance, delivery, and industrial inspections.

V. Summary and Q&A

- **Recap Key Points**:
  - AI and ML applications in computer engineering, from autonomous systems to predictive maintenance.
  - The importance of big data and cloud computing in processing vast amounts of information and enabling scalable solutions.
  - IoT's societal impact, including smart cities and healthcare innovations.
  - The growing role of robotics and automation in industry and the associated challenges.
- **Questions**: Open the floor for student questions.

References

Goodfellow, I., Bengio, Y., & Courville, A. (2020). *Deep Learning*. MIT Press.

Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

Hashem, I. A. T., et al. (2020). *The Rise of Big Data on Cloud Computing: Review and Open Research Issues*. Elsevier.

Marr, B. (2021). *Big Data in Practice: How 45 Successful Companies Used Big Data Analytics to Deliver Extraordinary Results*. Wiley.

Greengard, S. (2020). *The Internet of Things*. MIT Press.

Gubbi, J., et al. (2019). *Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions*. Elsevier.

Siciliano, B., & Khatib, O. (2021). *Springer Handbook of Robotics* (2nd ed.). Springer.

Craig, J. J. (2019). *Introduction to Robotics: Mechanics and Control* (4th ed.). Pearson.

MODULE 12-13

Technopreneurship in Computer Engineering

Learning Objectives:

By the end of this lesson, students will:

1. Understand the basics of technopreneurship in Computer Engineering.
2. Learn about different business models in the technology industry.
3. Gain insights into the culture of innovation and startups.
4. Recognize the role of engineers in technology-driven businesses.

I. **Basics of Technopreneurship in Computer Engineering**

1. **Introduction to Technopreneurship**:
   - **Definition**: Technopreneurship refers to entrepreneurship within the technology sector. It involves combining technological expertise with business acumen to create innovative products or services.
   - **Technopreneurs**: These are individuals who identify technological gaps or needs and create solutions by building companies or launching startups. In Computer Engineering, this often involves innovations in software, hardware, networking, or AI.
2. **Why Technopreneurship Matters**:
   - **Economic Growth**: Technology-driven businesses are often high-growth ventures that create jobs and drive economic development.
   - **Innovation**: Technopreneurship is a key driver of innovation, pushing boundaries in emerging fields like AI, IoT, and machine learning.
3. **Examples**:
   - Startups in **cloud computing** or **blockchain** technology often start with a technopreneur identifying a problem, then creating a technical solution and building a business around it (e.g., Dropbox, Ethereum).
4. **Activity**: Brief discussion where students share examples of technopreneurs they know or admire.

II. **Business Models in the Technology Industry**

1. **Introduction to Business Models**:
   - A **business model** describes how a company creates, delivers, and captures value. In the tech industry, different models cater to different products, services, and customer needs.

2. **Common Business Models in Technology**:
    - **SaaS (Software as a Service)**:
        - Companies provide software to customers over the internet on a subscription basis (e.g., Microsoft 365, Salesforce).
    - **Freemium**:
        - Basic services are provided for free, but users can pay for premium features (e.g., Spotify, LinkedIn).
    - **E-commerce**:
        - Companies sell products or services directly to consumers online (e.g., Amazon, Shopify).
    - **Platform Models**:
        - These connect different user groups, such as buyers and sellers (e.g., Uber, Airbnb), earning revenue through transaction fees or advertising.
3. **Revenue Streams in Tech**:
    - **Subscription**: Regular income from users who subscribe to a service (e.g., Netflix).
    - **Transaction Fees**: Small fees from each transaction, often used by platforms (e.g., PayPal, eBay).
    - **Advertising**: Companies generate revenue by displaying ads to users (e.g., Google, Facebook).
4. **Activity**: Students brainstorm a tech-related business idea and choose a suitable business model for it.

III. Innovation and Startup Culture (30 minutes)

1. **The Role of Innovation in Tech**:
    - **Definition of Innovation**: Innovation involves developing new ideas, products, or processes that solve existing problems in novel ways.
    - **Types of Innovation**:
        - **Product Innovation**: New or improved products (e.g., Tesla's electric cars).
        - **Process Innovation**: Improved methods of production or delivery (e.g., Amazon's automated warehouses).
        - **Business Model Innovation**: Introducing a new way to generate revenue or reach customers (e.g., Netflix shifting from DVDs to streaming).

2. **The Startup Ecosystem**:
    - **Startup Culture**: Startups are small, fast-moving companies focused on bringing innovative solutions to market. They often operate in dynamic environments where flexibility and rapid iteration are critical.
    - **Key Elements of Startup Culture**:
        - **Agility**: Startups must adapt quickly to market demands or technological changes.
        - **Risk-taking**: Entrepreneurs must be willing to take calculated risks to succeed in a competitive market.
        - **Collaboration**: Teams often work in open, collaborative environments that foster creativity.
3. **Challenges for Startups**:
    - **Funding**: Raising capital is one of the biggest challenges. Many rely on venture capital, angel investors, or crowdfunding.
    - **Competition**: The technology sector is highly competitive, and companies must differentiate themselves through unique value propositions.
4. **Activity**: A brief class discussion on how innovation and startup culture contribute to advancements in Computer Engineering.

IV. Role of Engineers in Technology-Driven Businesses

1. **Engineers as Innovators**:
    - Engineers are often at the core of technological innovation. Their technical expertise allows them to solve complex problems and develop new solutions in fields like software development, hardware design, and network systems.
    - **Entrepreneurial Engineers**: Many engineers become technopreneurs, starting companies that focus on technology-based products or services. Famous examples include Elon Musk (Tesla, SpaceX) and Bill Gates (Microsoft).

2. **Collaboration Between Engineers and Business Leaders**:
    - Engineers work closely with business teams to ensure that technological solutions align with market needs. They must understand both the technical aspects and the business implications of their work.
3. **Skillsets Engineers Bring to the Table**:
    - **Problem-solving**: Engineers are trained to approach problems methodically, using data and logic to find solutions.

- **Innovation**: Their deep understanding of technology enables them to push the boundaries of what is possible.
- **Leadership in Tech Companies**: Many engineers take on leadership roles in technology companies, guiding teams through product development and innovation (e.g., Sundar Pichai at Google).

4. **Ethical Responsibilities**:
   - Engineers must also be aware of their ethical responsibilities, ensuring that their innovations do not harm society or the environment.
5. **Activity**: Ask students to identify one role that engineers play in technology-driven businesses and share examples of engineers who have contributed to major technology advancements.

V. Conclusion (5 minutes)

1. **Recap**:
   - Technopreneurship plays a critical role in the development of the tech industry.
   - Business models in technology vary based on products, services, and customer interactions.
   - Innovation and startup culture are central to the tech industry's growth and evolution.
   - Engineers are key players in both technological innovation and business development.
2. **Q&A**:
   - Open the floor for questions or additional thoughts on any of the topics discussed.

References

Damodaran, A. (2021). *The Business of Tech Startups: How Business Models Shape Innovation*. Wiley.

Bailetti, T. (2020). *Technopreneurship: A Practical Approach for Startups and Entrepreneurs in the Technology Industry*. Springer.

Schilling, M. (2022). *Strategic Management of Technological Innovation*. McGraw-Hill Education.

Ries, E. (2019). *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business.

Griffin, R. (2023). *Entrepreneurial Engineers: How Technological Innovation Drives Business*. CRC Press.

MODULE 14-15

Ethics and Professional Responsibility

Learning Objectives:

By the end of this lesson, students will be able to:

- Understand the key ethical issues in computing and engineering.
- Recognize the importance of legal aspects and intellectual property rights in engineering.
- Identify the role of professional organizations and certifications for Computer Engineers.

I. **Ethical Issues in Computing and Engineering**

1. **Introduction to Ethics in Computing**:
    - **Ethics** refers to a set of moral principles that govern behavior. In the field of computing and engineering, it involves making decisions that protect public safety, privacy, and welfare.
    - Ethical concerns arise from the impact of technology on society and the environment.
2. **Common Ethical Issues in Computing**:
    - **Privacy and Data Security**: Engineers must ensure that personal data is handled securely, protecting users from breaches and misuse.
    - **Artificial Intelligence (AI) and Automation**: Ethical concerns about job displacement, decision-making by AI, and the potential for bias in AI systems.
    - **Cybersecurity**: Ethical responsibilities in designing secure systems to prevent hacking, data theft, and cyberattacks.
    - **Environmental Impact**: The sustainability of computing systems, especially with the rising energy demands of data centers and electronic waste.
3. **Examples of Ethical Dilemmas**:
    - **Case Study**: The Cambridge Analytica scandal, where unethical use of personal data on Facebook influenced political campaigns, raising concerns about user privacy and consent.
    - **Software Piracy**: The illegal copying and distribution of software, which raises ethical and legal concerns.

4. **Ethical Frameworks**:
    - **Utilitarianism**: The principle of maximizing the overall good or minimizing harm.
    - **Deontology**: Following ethical rules and duties regardless of the outcome.
    - **Virtue Ethics**: Focusing on the moral character of individuals rather than rules or consequences.
5. **Professional Codes of Ethics**:
    - Organizations like the **IEEE** (Institute of Electrical and Electronics Engineers) and the **ACM** (Association for Computing Machinery) have established codes of ethics to guide engineers in making ethical decisions.
    - **Key Principles**: Integrity, responsibility, fairness, and respect for privacy.

II. Legal Aspects and Intellectual Property Rights

1. **Introduction to Legal Aspects in Engineering**:
    - Engineers must understand and comply with relevant laws and regulations that govern their work. These laws ensure that technology is developed responsibly and fairly.
2. **Key Legal Issues in Computing**:
    - **Data Protection Laws**: Regulations like the **General Data Protection Regulation (GDPR)** in the EU protect the privacy of individuals by controlling how companies collect and use personal data.
    - **Software Licensing**: Legal frameworks that dictate how software can be used, distributed, or modified (e.g., open-source vs. proprietary software).
    - **Cybercrime and Digital Laws**: Laws addressing online activities such as hacking, identity theft, and the distribution of malicious software.
3. **Intellectual Property Rights (IPR)**:
    - **Intellectual Property (IP)** refers to the creations of the mind, such as inventions, literary works, and symbols used in commerce.
    - **Types of Intellectual Property Rights**:
        - **Patents**: Protect inventions and grant the holder exclusive rights to use, sell, or license the invention for a period of time.
        - **Copyrights**: Protect original works of authorship, such as software code, written documents, and designs.
        - **Trademarks**: Protect brand names, logos, and symbols associated with products or services.
        - **Trade Secrets**: Protect confidential business information from being disclosed or used by competitors.

4. **Case Study**:
    - **Apple vs. Samsung Patent Dispute**: A legal battle over patent infringement, where both companies claimed violations of intellectual property rights related to smartphone design and technology.
5. **Software Licensing**:
    - **Open Source Licenses**: Allow users to modify and distribute software under specific conditions (e.g., GNU General Public License).
    - **Proprietary Licenses**: Restrict usage, distribution, and modification rights to the creator or company.

III. Professional Organizations and Certifications for Computer Engineers (30 minutes)

1. **Importance of Professional Organizations**:
    - Professional organizations offer **networking opportunities**, **career development**, and **access to industry knowledge**. They also set ethical standards and provide certifications that validate skills and expertise.
2. **Key Professional Organizations for Computer Engineers**:
    - **IEEE (Institute of Electrical and Electronics Engineers)**:
        - The world's largest technical professional organization dedicated to advancing technology. IEEE offers resources for engineers in electronics, telecommunications, and computer science.
        - Membership benefits include access to conferences, publications, and certification programs.
    - **ACM (Association for Computing Machinery)**:
        - A leading organization for computing professionals, ACM promotes computing research, education, and ethical practice.
        - Offers special interest groups (SIGs) focused on various areas like AI, software engineering, and human-computer interaction.
3. **Certifications for Computer Engineers**:
    - **Certified Information Systems Security Professional (CISSP)**:
        - A globally recognized certification in cybersecurity, which demonstrates expertise in designing and managing security programs.
    - **Cisco Certified Network Associate (CCNA)**:
        - A certification that validates skills in networking fundamentals, IP connectivity, and network access.
    - **Project Management Professional (PMP)**:

- A certification that is valuable for engineers involved in managing large-scale technology projects, focusing on project leadership and execution.

4. **Advantages of Certifications**:
    - Enhance career prospects by demonstrating expertise and dedication to ongoing professional development.
    - Stay updated with the latest trends, tools, and best practices in the industry.

IV. **Summary and Q&A (10 minutes)**

- **Recap Key Points**:
    - Ethical issues in computing such as privacy, AI ethics, and data security are critical for engineers to address.
    - Legal aspects such as intellectual property rights and data protection laws are necessary to ensure responsible technological development.
    - Professional organizations like **IEEE** and **ACM** provide engineers with the tools and knowledge to thrive in their careers, and certifications can enhance their professional growth.
- **Questions**: Engage students in an open discussion about ethical dilemmas they might encounter in their future careers.

References

Baase, S., & Henry, T. (2019). *A Gift of Fire: Social, Legal, and Ethical Issues for Computing Technology* (5th ed.). Pearson.

Patterson, M. (2021). *Intellectual Property Law in the Digital Age* (3rd ed.). Oxford University Press.

IEEE. (2023). *IEEE Code of Ethics and Professional Conduct*. IEEE Press.

Reynolds, G. (2020). *Ethics in Information Technology* (7th ed.). Cengage Learning.

Stallings, W. (2021). *Computer Security: Principles and Practice* (4th ed.). Pearson.

Halbert, T., & Ingulli, E. (2020). *Law & Ethics in the Business Environment* (9th ed.). Cengage Learning.

MODULE 16-17

CPE Practice and Design Discussion and Future Outlook

Learning Objectives:

By the end of this lesson, students will:

- Understand the future challenges and opportunities in Computer Engineering.
- Recognize the role of CPE (Computer Engineering) Practice and Design in the curriculum.
- Learn how to prepare for research in Computer Engineering.
- Know the expectations for final assessments and project submissions.

I. Future Challenges and Opportunities in Computer Engineering (40 minutes)

1. **Introduction**:
    - Computer Engineering is a constantly evolving field, bringing both challenges and exciting opportunities. This section will explore some of these.
2. **Key Challenges**:
    - **Cybersecurity Threats**:
        - As systems grow more interconnected (cloud computing, IoT), ensuring secure data transmission and storage becomes critical.
    - **Sustainability**:
        - Energy consumption in data centers and personal devices is a growing concern. Future engineers must focus on creating energy-efficient technologies.
    - **Ethical Issues**:
        - With AI and automation impacting job markets and society, ethical considerations surrounding these technologies will be vital.
3. **Emerging Opportunities**:
    - **Artificial Intelligence (AI)**:
        - AI is transforming industries such as healthcare, finance, and manufacturing. Computer Engineers will play a key role in developing AI-driven solutions.
    - **5G and IoT**:
        - The expansion of 5G and the Internet of Things (IoT) provides new opportunities for smart cities, automation, and real-time data processing.

- **Quantum Computing**:
    - While still in its early stages, quantum computing promises to solve complex problems that current systems cannot handle, offering great future potential.

4. **Class Activity (Discussion)**:
    - Break students into groups and assign each group a current technological challenge (e.g., cybersecurity, energy efficiency, or quantum computing). Ask them to discuss potential solutions and share their ideas with the class.

II. CPE Practice and Design Introduction and Its Role in the Curriculum

1. **What is CPE Practice and Design?**
    - CPE Practice and Design is a core part of the Computer Engineering curriculum where students apply theoretical knowledge to real-world engineering problems.
    - It encompasses both hardware (e.g., circuit design) and software (e.g., system development) aspects, providing hands-on experience.
2. **Goals of CPE Practice and Design**:
    - Develop practical skills in system design, debugging, and testing.
    - Foster critical thinking, problem-solving, and creativity in tackling engineering challenges.
    - Encourage collaboration and teamwork, as most projects are conducted in groups.
3. **Stages of a Design Project**:
    - **Conceptualization**:
        - Identifying a problem or challenge to solve, proposing innovative solutions, and defining system requirements.
    - **Design and Development**:
        - Creating schematics, writing code, and assembling components.
    - **Testing and Evaluation**:
        - Debugging, running tests to ensure functionality, and refining the project based on results.
4. **Role in Curriculum**:
    - CPE Practice and Design projects often take place in the final year, providing a capstone experience where students synthesize knowledge gained throughout their studies.

III. Preparing for Research in Computer Engineering

1. **Introduction to Research**:
    - Research is a key component of Computer Engineering, pushing the boundaries of what technology can achieve.
    - Research helps solve unanswered questions, improves existing technology, and can lead to innovative breakthroughs.
2. **Key Research Areas**:
    - **Machine Learning**: Exploring how machines can learn and adapt to new data.
    - **Embedded Systems**: Developing real-time systems with small, specialized computing devices.
    - **Cyber-Physical Systems**: Integrating physical systems with computational power (e.g., robotics).
3. **Steps to Conduct Research**:
    - **Literature Review**:
        - Start by reviewing academic journals, books, and papers in the area of interest. Identify research gaps that your work can address.
    - **Formulate Research Question**:
        - Define a clear and specific research question or hypothesis based on the review.
    - **Methodology**:
        - Choose appropriate methods (experiments, simulations) to explore the problem.
    - **Data Collection**:
        - Gather data from simulations, prototypes, or tests, and analyze the results to draw conclusions.
4. **Research Proposal**:
    - Students preparing for their capstone or research project must submit a proposal outlining their objectives, methodology, and expected outcomes.

IV. Final Assessment and Project Submission

1. **Overview of the Final Assessment**:
    - The final assessment will evaluate students' understanding of core concepts and their ability to apply them to practical problems.
    - The assessment is divided into two parts:
        - **Written Exam**: Testing theoretical understanding.

- **Capstone/Project Submission**: Evaluating the practical skills demonstrated in the capstone project.

2. **Key Components of the Final Project**:
   - **Innovation and Creativity**: Students must showcase how their project offers a novel solution to a real-world problem.
   - **Technical Merit**: Projects will be evaluated on the technical quality of the design, development, and implementation.
   - **Documentation**: Clear and thorough documentation, including a detailed report outlining design decisions, development stages, testing, and results.
   - **Presentation**: Students must present their projects to faculty, explaining their design process and demonstrating the functionality.
3. **Project Submission Guidelines**:
   - Ensure all documentation, source code, and any necessary hardware are submitted by the deadline.
   - A final presentation of the project will be required, followed by a Q&A session.

V. Conclusion and Summary (10 minutes)

1. **Recap**:
   - The future of Computer Engineering holds many challenges and opportunities, particularly in areas like cybersecurity, AI, and quantum computing.
   - CPE Practice and Design is a crucial part of the curriculum, allowing students to apply their learning to real-world projects.
   - Preparing for research involves careful planning, including literature reviews, defining research questions, and selecting appropriate methodologies.
   - The final assessment and project submission are key components of your academic evaluation and will test both your theoretical knowledge and practical skills.
2. **Q&A**:
   - Open the floor for questions and clarifications on any topics covered during the lesson.

References

Anderson, J. (2022). *Future Trends and Challenges in Computer Engineering*. CRC Press.

Miller, R. (2021). *Design and Practice in Computer Engineering: A Student's Guide*. Springer.

Garcia, M. (2023). *Research Methods for Computer Engineers: Techniques and Applications*. Pearson.

Hughes, B. (2020). *Capstone Projects and Assessment in Engineering Education*. Wiley.

Thompson, L. (2019). *Ethical Considerations in Computing and Engineering Research*. MIT Press.

RECOMMENDATION FOR CERTIFICATION TO THE UTLDO DIRECTOR

(Form 3)

September 21, 2024
Date

PROF. ELMER DE JOSE, PhD
Director
University Teaching and Learning Development Office
This University

Dear Dir. De Jose:

The instructional material(s) listed on the attached page with their corresponding authors/developers, points earned, and length of validity have already passed our final evaluation. Accordingly, the said materials can be issued with their respective certifications.

Thank you for your consideration.

Sincerely,

ENGR. ROLITO L. MAHAGUAY, PCPE
Overall Committee Chair

LIST OF INSTRUCTIONAL MATERIALS RECOMMENDED FOR CERTIFICATION

(Form 4)

Please check (✓) the type of the following Instructional Materials: Learning Module: ☑ Video Recording: ☐

No.	App No.	Writers/Developers	College/Branch/ Campus	Title of IM	Points Earned	Validity
1.		Engr. Rolito L. Mahaguay	CEA/Mabini	Computer Engineering as a Discipline	95	2
2.		Engr. Pedrito M. Tenerife, Jr.	CEA/Mabini	Computer Engineering as a Discipline	95	2
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						