# Impact of GenAI on Higher Education

Pankaj Jalote
Distinguished Professor (and Founding Director)
IIIT - Delhi

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
**DELHI**

# Background

- GenAI has taken the world by a storm – it is clearly expected to impact many aspects of knowledge-based production
- Due to its generative capability (and easy availability) it is likely to impact HE significantly (actually will impact all education) as process of education requires students to generate some outputs (solutions, reports, code, presentations, proofs, models, …)
- We will discuss how it may impact HE – consider different levels in higher education and explore how it may impact at that level
- Based on personal views and discussions

# Learning in HE

- Higher education: it takes school students and produces graduates with some desired knowledge, skills, dispositions
  - HE is about students' learning – teaching guides their learning
- Can GenAI really impact learning, e.g. can it make learning faster/more efficient or increase learning capacity? Not clear!
- GenAI can impact the goals of learning (what to learn) and the overall education process – will discuss some possibilities

# Higher Education

- HE designs various programs (BA, Btech… in X, Y, …)
- Defining characteristics (design) of a program
  1. Graduate attributes the program aims to develop
  2. Program – network of courses to develop those attributes
  3. Course design – defined by learning outcomes of the course
  4. Teaching of a course
- GenAI is already impacting teaching (4) – its impact on higher aspects (3, 2, 1) of education will come

# Graduate Attributes

- A program has some some general graduate attributes – these are General GA s common across all BTechs
- Also has discipline specific GA s – these specify discipline specific attributes
- Together the GAs define the goal of the program; how much general and how much discipline, is a balancing choice

# Graduate Attributes (GAs)

**Computer Science GAs (Examples)**

1. Proficiency in writing programs …
2. Design and apply appropriate algos and data structures for solving problems..
3. Understand computing systems at different levels – CO, OS, NW…
4. Understand foundations, principles, limits of computing
5. Analyse data for learning, prediction, decision making, etc.
6. …

**General GAs (Examples)**

1. Ability to identify, analyse and evolve approaches for solving problems
2. Ability to apply mathematical concepts and techniques in problem solving
3. Ability to function in teams
4. Ability to communicate effectively..
5. Ability to self-learn and engage in life-long learning…
6. An understanding of professional and ethical responsibility

# Program

- A program is a network of credit units – courses, internships, practice, projects, etc towards some goals
- Curriculum should aim to develop the graduate attributes
  - Constraints: have 8 semesters, each with about 5 courses
  - There is only so much that can be absorbed in this box
- Courses: some common core, some discipline core, some discipline electives, some open electives

# Example: CS Core (AICTE, 2022)

- Data structures and Algorithms
- Discrete Mathematics
- Computer Organization and Architecture
- Advanced Programming (in lieu of OOP)
- Operating Systems
- Design and Analysis of Algorithms
- Database Systems
- Computer Networks
- Machine Learning
- Security

# Course Design

- A good course design has
  A. Learning outcomes (goals)
  B. Teaching and learning activities, e.g. lecturing, tutorials, …
  C. Assessment and feedback

- These three are related

# Learning Outcomes

- Statements on understanding and skill of students at the end of the course
- E.g. Intro to Programming Course (Python) LO s:
  1. Be fluent in the use of procedural statements — assignments, conditional statements, loops.
  2. Effectively use built-in data structures like lists, dictionaries, sets
  3. Be able to design, code, and test small Python programs that meet requirements expressed in English

# Teaching (and Assessment)

- Teaching – lectures, tutorials, give assignments, exams, projects, …
- Assessment – to evaluate, measure, and document the academic readiness, learning progress… of students; Common instruments
  - Assignments
  - Projects
  - Quizzes, tests, exams
  - …
- Assessments help students learn what was taught – an integral part of teaching

# GenAI and Teaching

- Assessments have gotten harder – e.g. how do you get students to do practice exercises with genAI available
  - Most assessments ask students to generate something – so genAI can be used by students
  - Much of HE is currently struggling with this
  - Effective policies, approaches, etc. will evolve

- GenAI can provide personalized assistance – which instructors/TAs cannot provide (due to scale)
  - Work happening here (e.g. BotTA)
- Can help instructors in preparing lectures, examples, assignments
  - Some instructors already using for lectures
  - Preparing assessments ??

# GenAI and Learning Outcomes

- If genAI is used as a tool by students, some learning outcomes may become redundant
  - Eg. In intro to programming, students can “write” simple modules using genAI – is this redundant??
- What should be the new learning outcomes for a course? Debate will reach there (has not yet reached)
  - Eg. In intro to programming, if students can “generate” simple modules using genAI, what should the course now teach?
  - Q: Can a student learn higher level concepts without being proficient in programming (traditional LOs)?
- Courses’ LOs will evolve when genAI is accepted as a tool to solving more complex problems, working at higher levels of abstraction

# Program Design

- New courses (mostly as electives) will be added in programs
  - On the technology and its capability
  - Already beginning to happen
- Core may become more compact

# Impact on Graduate Attributes

- The debate will reach here later, after the impact of genAI on various aspects is clearer

- Not clear how graduate attributes will / should change – should the graduate attributes include something like “Proficient in use of LLMs for various tasks….”
  - If so, this has major implications on design of courses

# Some Initial Thoughts on CS Programs

- We have different courses covering different aspects of CS – programming, data structures, algorithms, OS, databases, …
  - To prepare students to take up industry careers or research career
- Our focus is generally on helping students understand some concepts, gain some knowledge, and develop some skills
  - Concepts: To build a vocabulary of concepts based on which new concepts are learned
  - Knowledge: To provide knowledge that can be useful for student for problem solving
  - Skill: What students can do
- With LLMs what concepts to teach, what knowledge to impart, and what skills to develop will be impacted

# Concepts and Knowledge

- Concepts in each topic will remain important – they provide the vocabulary for thinking and further learning and building new knowledge
  - E.g. computational thinking; programming concepts; concept of data structure and algorithms, efficiency and complexity of algorithms, correctness of algorithms, limits of computation, paradigms; concept of databases and query language; system related concepts – how OS work, how programs run; …
  - Concepts cannot be understood without understanding some examples (knowledge) of those concepts (e.g. some actual algorithms)
- With LLMs some knowledge we teach for possible use will be less relevant
  - E.g. some algorithms, SQL, programming different algorithms, …
- So, some of the knowledge being taught may be omitted, while keeping those that develop conceptual understanding and builds foundations

# Skills

- Some skills may become irrelevant and some new skills may become important – which become irrelevant (e.g. SQL) will evolve
- Some higher level skills will become more relevant – not clear what these are
- E.g. Developing a useful software is now not about writing stand-alone programs (which CS education focuses on); the complexity is shifting to composing an application using libraries, frameworks, databases, cloud, external services
  - Developing concepts, knowledge, and skills on this will definitely be more important in coming years
  - CS Education largely ignores this complexity

# Courses in CS Programs

- If core focuses on concepts only and knowledge component is reduced, this can free up hours
- What new courses will fill the available slots - not clear
  - Advanced courses on topics?
  - More practice on applying concepts and leveraging available knowledge?
  - More focus on society, ethics, and profession – as this becomes more important; how to teach this is a challenge?
  - More application courses for different domains?

# Summary

- HE is about learning and teaching
- Impact of AI on learning by students – not clear
- AI can impact teaching – has already affected assessment
- Will affect: learning outcomes of courses and graduate attributes, leading to redesign of programs
- Shared some thoughts on impact on CS programs – on concepts and knowledge, and skill development