

Question 1: Provide an illustration on the advancement of computing over the last 2 decades

Answer: Computers truly came into their own as great inventions in the last two decades of the 20th century. But their history stretches back more than 2500 years to the abacus: a simple calculator made from beads and wires, which is still used in some parts of the world today. The difference between an ancient abacus and a modern computer seems vast, but the principle—making repeated calculations more quickly than the human brain—is exactly the same. Only 20 years ago, personal computers are becoming sufficiently affordable for families to own, but in today's society it's not uncommon to spot primary school children with smartphones packed with countless apps which seem to do every single thing imaginable. The internet has been easier to access than ever: with one click of a button one can see and hear their friend or a relative on the other side of the world and send them a virtual chicken. There are (contentious) narratives of ever-improving technology, but also an entirely unarguable narrative of computing itself being transformed by becoming so deeply embedded in our daily lives. This is, in many ways, the essence of big data. Computing is being fed by human data streams: traffic data, airline trips, banking transactions, social media and so on. The challenges of the discipline have been dramatically changed by this data, and also by the fact that the products of the data (such as traffic control and targeted marketing) have immediate impacts on people. Software that runs robustly on a single computer is very different from that with a high degree of rapid interaction with the human world, giving rise to needs for new kinds of technologies and experts, in ways not even remotely anticipated by the researchers who created the technologies that led to this transformation. Decisions that were once made by hand-coded algorithms are now made entirely by learning from data. Whole fields of study may become obsolete.

Question 2: What is the difference between R and Python?

Answer:

Library repository

R has an exhaustive repository of libraries that are regularly updated in CRAN which is Comprehensive R Archive Network. It includes dplyr, mice, ggplot2 and many more.

Python includes a pip package index that has all the libraries for Python. They include matplotlib, pandas, seaborn and many more.

Data handling

R has the power of doing data analysis without loading any package in its memory. There are many packages that are available in R that can be utilized to study data and get insights from them.

Python, we need packages like numpy and pandas for creating a data frame and processing the data.

Statistical Correctness

Since R is developed for data statistics, hence it provides better support and libraries for statistics.

Python is best used for application development and deployment. But R and its libraries implement a wide variety of statistical and graphical techniques for data analysis.

Ease of Learning

R has a steep learning curve and people with less or no experience in programming finds it difficult in the beginning. Once you get a grip of the language, it is not that hard to understand.

Python emphasizes on productivity and code readability which makes it one of the simplest programming languages. It is preferable due to its ease of learning and understandability.

Speed

R is a low-level programming language due to which it requires longer codes for simple procedures. This is one reason for the reduced speed.

Python is a high-level programming language and it has been the choice for building critical yet fast applications.

Question 3: What is the difference between Functional and Event-based programming?

Answer - In computer science, functional programming is a programming paradigm—a style of building the structure and elements of computer programs—that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data. It is a declarative programming paradigm in that programming is done with expressions or declarations^[1] instead of statements.

In computer programming, event-driven programming is a programming paradigm in which the flow of the program is determined by events such as user actions (mouse clicks, key presses), sensor outputs, or messages from other programs or threads. Event-driven programming is the dominant paradigm used in graphical user interfaces and other applications (e.g., JavaScript web applications) that are centered on performing certain actions in response to user input.

Question 4: What do you understand to be ethical hacking?

Ethical hacking and ethical hacker are terms used to describe hacking performed by a company or individual to help identify potential threats on a computer or network. An ethical hacker attempts to bypass system security and search for any weak points

that could be exploited by malicious hackers. This information is then used by the organization to improve the system security, to minimize or eliminate any potential attacks.

For hacking to be deemed ethical, the hacker must obey the following rules:

- Expressed (often written) permission to probe the network and attempt to identify potential security risks.
- You respect the individual's or company's privacy.
- You close out your work, not leaving anything open for you or someone else to exploit at a later time.
- You let the software developer or hardware manufacturer know of any security vulnerabilities you locate in their software or hardware, if not already known by the company.