

Directions:

Read the following four texts. Answer the questions after each text by choosing A, B, C or D. Mark your answers on the **ANSWER SHEET**. (40 points)

Text 1

It's true that high-school coding classes aren't essential for learning computer science in college. Students without experience can catch up after a few introductory courses, said Tom Cortina, the assistant dean at Carnegie Mellon's School of Computer Science.

However, Cortina said, early exposure is beneficial. When younger kids learn computer science, they learn that it's not just a confusing, endless string of letters and numbers – but a tool to build apps, or create artwork, or test hypotheses. It's not as hard for them to transform their thought processes as it is for older students. Breaking down problems into bite-sized chunks and using code to solve them becomes normal. Giving more children this training could increase the number of people interested in the field and help fill the jobs gap, Cortina said.

Students also benefit from learning something about coding before they get to college, where introductory computer-science classes are packed to the brim, which can drive the less-experienced or less-determined students away.

The Flatiron School, where people pay to learn programming, started as one of the many coding bootcamps that have become popular for adults looking for a career change. The high-schoolers get the same curriculum, but "we try to gear lessons toward things they're interested in," said Victoria Friedman, an instructor. For instance, one of the apps the students are developing suggests movies based on your mood.

The students in the Flatiron class probably won't drop out of high school and build the next Facebook. Programming languages have a quick turnover, so the "Ruby on Rails" language they learned may not even be relevant by the time they enter the job market. But the skills they learn – how to think logically through a problem and organize the results – apply to any coding language, said Deborah Seehorn, an education consultant for the state of North Carolina.

Indeed, the Flatiron students might not go into IT at all. But creating a future army of coders is not the sole purpose of the classes. These kids are going to be surrounded by computers – in their pockets, in their offices, in their homes – for the rest of their lives. The younger they learn how computers think, how to coax the machine into producing what they want – the earlier they learn that they have the power to do that – the better.

21. Cortina holds that early exposure to computer science makes it easier to_____.

- [A] complete future job training
- [B] remodel the way of thinking
- [C] formulate logical hypotheses
- [D] perfect artwork production

22. In delivering lessons for high-schoolers, Flatiron has considered their_____.

- [A] experience
- [B] interest
- [C] career prospects
- [D] academic backgrounds

23. Deborah Seehorn believes that the skills learned at Flatiron will_____.

- [A] help students learn other computer languages
- [B] have to be upgraded when new technologies come
- [C] need improving when students look for jobs
- [D] enable students to make big quick money

24. According to the last paragraph, Flatiron students are expected to_____.

- [A] bring forth innovative computer technologies
- [B] stay longer in the information technology industry
- [C] become better prepared for the digitalized world
- [D] compete with a future army of programmers

25. The word "coax" (Line 4, Para. 6) is closest in meaning to_____.

- [A] persuade
- [B] frighten
- [C] misguide
- [D] challenge