**Using AI Chatbots: A Guide to the Perplexed**

**שימוש בסוכני בינה מלאכותית: מורה נבוכים ונבוכות**

By “AI Chatbots” we refer to such programs as *ChatGPT* (by Open AI), *Claude* (by Anthropic), *Copilot* (by Microsoft), and similar tools. An “AI chatbot” can be described as a conversational search engine that simulates an intelligent, personal assistant.

Chatbots can greatly enhance your learning experience. In particular, in this semester we will launch our own chatbot, which we call RUNIbot, or Rbot for short. Rbot is designed specifically for students in RUNI’s Introduction to Computer Science course. When you ask Rbot a question, it bases its answer *first* on the materials that were already learned in the course, and only then on materials that can be found elsewhere, in similar courses and on the Internet at large. Therefore, interacting with Rbot provides specific information and guidance which is focused and relevant to our Intro to CS course.

That said, you are not restricted to using Rbot. You can use any other chatbots, or not use them at all. If you choose to use chatbots, it’s important to do so wisely and responsibly. Over-reliance on chatbots can diminish your ability to think independently and use information critically. Specifically in your journey as a computer science student, becoming too dependent on chatbots can inhibit your performance in two mission-critical situations: *exams* and *job interviews*. In both situations, you won’t have access to chatbots or to the Internet. Therefore, to succeed in this course and beyond, it’s essential to develop skills to solve problems and write code independently.

So, how can you use chatbots wisely, to enhance your learning performance and career prospects? That’s what this little document is all about.

**1. Use AI for research and clarification, not shortcuts**

Use chatbots for answering questions and providing explanations. Don’t use them as a shortcut to avoid doing the hard work yourself. For example, if you are working on some programming or algorithmic problem, ask the chatbot to explain the underlying principles, not to solve the problem for you. Understanding the *why* and *how* is far more important than getting to the right answer. For example, suppose that Bob and Alice were asked to write a program that performs binary search. Bob prompted the chatbot to “*Write a Java program that performs binary search*”. Alice’s prompt was “*Explain binary search, and give an example*”. Bob will undermine his ability to write code without external help. Alice will be motivated and guided to solve the problem on her own.

**2. Verify information**

AI chatbots can provide incorrect or incomplete information. Therefore, develop a habit to double-check what the chatbot says. To do so, confirm the chatbot’s answers using your course materials and common sense. For example, after asking "*How can I check if two strings are equal in Java*?", a chatbot answered: ”*You can compare the two strings using the* == *operator*”. As it turns out, this answer sometimes works, but is generally incorrect. If the question will be given to Rbot, the chatbot is more likely to give the correct answer. That’s because Rbot is trained to first use our Intro to CS course materials, and correct string comparison happens to be something that we discuss and demonstrate in many examples in this course.

**3. Ask clear and specific questions**

For example, instead of asking "*What is a loop*?" ask "*Can you explain the difference between a* for *loop and a* while *loop*?" This will lead to more detailed and useful explanations, helping you gain deeper insights into the material. And if you use Rbot, you will get more focused answers, using examples from the course.

**4. Don't rely on AI for problem-solving**

When it comes to programming or solving algorithmic problems, it's important to **practice handling these problems on your own**. Deep learning happens only when you struggle with a problem, make mistakes, and debug your code by trial and error. To reiterate an important point: In your exams and job interviews, you will not have access to AI assistance. Therefore, you must develop your ability to approach and solve problems on your own.

**5. Use AI to supplement, not replace, your learning**

Chatbots can supplement your learning experience, but they don’t replace traditional learning methods. Attending lectures, and doing homework assignments on your own, are vital components of a strong computer science foundation. Use chatbots as an additional resource to clarify doubts and fill in missing information, but keep your primary focus on learning through direct engagement with the course materials. In particular, go over the lecture slides after each lecture, play with the given code examples, and, optionally, use a chatbot to enhance your learning. For example, you can copy-paste a code example that was given in the lecture, and ask the chatbot to explain, or improve it. Importantly, Rbot is programmed to use only materials that were learned up to the current week in the course; Therefore, when answering questions, it will not use advanced concepts that were not learned yet (unless you specifically ask it to do so).

**6. Learn to think critically and creatively**

One of the key skills you will develop in computer science is critical thinking. While chatbots are great at providing answers, they don’t teach you how to **ask the right questions** or think creatively. Push yourself to explore multiple solutions to a problem and consider alternative approaches. This kind of thinking will serve you well not only in exams but also in your future career as a computer scientist. For example, if you wrote code that solves a problem, and the code works correctly, you can copy-paste the code into the chatbot, and ask it to show you other ways to solve the problem. Seeing multiple solutions and understanding their pros and cons provides a tremendous educational value.

**7. Use old school learning techniques**

When you sit for job interviews, recruiters will assess your problem-solving skills and knowledge, not your ability to use AI tools. Focus on understanding the core concepts thoroughly so that you can demonstrate your skills confidently without external assistance. Practicing problems by hand, using paper and pencil, writing on whiteboards, coding from scratch in pseudo-code –– these are the best techniques for preparing for exams and job interviews.