

**You said:**

web development, app development, cyber security...without these, what else can i become studying from CSE and get a 7 figure

**ChatGPT said:**

ChatGPT

If you're studying Computer Science and Engineering (CSE) and want to explore high-paying career paths beyond web development, app development, and cybersecurity, here are some lucrative options to consider:

**1. Data Science and Machine Learning**

* **Role**: Data Scientist, Machine Learning Engineer, AI Specialist
* **Why it's lucrative**: These professionals work on building intelligent systems and extracting insights from large datasets, which is critical for many industries.
* **Skills**: Python, R, SQL, TensorFlow, PyTorch, data visualization.
* **Potential Earnings**: Often reaches 7 figures annually with experience and specialization.

**2. Cloud Computing**

* **Role**: Cloud Architect, DevOps Engineer, Cloud Security Specialist
* **Why it's lucrative**: Companies rely on scalable, secure, and cost-efficient cloud solutions, and skilled professionals are in high demand.
* **Skills**: AWS, Azure, Google Cloud Platform, Docker, Kubernetes.
* **Potential Earnings**: Experienced architects and engineers command high salaries.

**3. Blockchain Development**

* **Role**: Blockchain Developer, Smart Contract Developer
* **Why it's lucrative**: The growing popularity of cryptocurrencies and decentralized applications creates demand for blockchain experts.
* **Skills**: Solidity, Ethereum, Hyperledger, cryptography.
* **Potential Earnings**: 7 figures, especially in decentralized finance (DeFi) and Web3 projects.

**4. Embedded Systems and IoT**

* **Role**: IoT Engineer, Embedded Software Developer
* **Why it's lucrative**: The rise of connected devices and automation creates demand for specialists in embedded systems and IoT.
* **Skills**: C/C++, Python, microcontrollers, Arduino, Raspberry Pi.
* **Potential Earnings**: Strong prospects in industries like healthcare, automotive, and smart devices.

**5. Quantitative Analysis**

* **Role**: Quant Developer, Algorithmic Trader
* **Why it's lucrative**: Working in finance, quant analysts build models to predict market trends and automate trades.
* **Skills**: Python, R, C++, financial modeling, statistics.
* **Potential Earnings**: Extremely high, especially in hedge funds and investment banking.

**6. Computer Graphics and Game Development**

* **Role**: Game Developer, Graphics Programmer, Simulation Expert
* **Why it's lucrative**: Entertainment and gaming industries are booming, offering lucrative opportunities for skilled developers.
* **Skills**: Unity, Unreal Engine, C++, OpenGL, DirectX.
* **Potential Earnings**: Highly competitive salaries in AAA gaming studios.

**7. AI and Robotics**

* **Role**: Robotics Engineer, AI Specialist
* **Why it's lucrative**: Automation and robotics are reshaping industries from manufacturing to healthcare.
* **Skills**: ROS (Robot Operating System), MATLAB, Python, AI/ML integration.
* **Potential Earnings**: High-paying roles in innovative tech fields.

**8. Research and Academia**

* **Role**: Research Scientist, Professor
* **Why it's lucrative**: Specialized knowledge in areas like quantum computing, bioinformatics, or theoretical computer science can lead to prestigious and high-paying positions.
* **Skills**: Deep technical expertise, publications, teaching.
* **Potential Earnings**: Research roles in elite institutions or collaborations with tech giants can be rewarding.

**9. Product Management in Tech**

* **Role**: Product Manager, Tech Consultant
* **Why it's lucrative**: Bridging the gap between technology and business, these roles oversee development and delivery of tech products.
* **Skills**: Technical understanding, project management, communication.
* **Potential Earnings**: High-paying, especially in software or hardware companies.

**10. Ethical Hacking and Digital Forensics**

* **Role**: Ethical Hacker, Forensic Analyst
* **Why it's lucrative**: A subset of cybersecurity, these roles are essential for organizations looking to safeguard against digital threats.
* **Skills**: CEH, penetration testing, digital investigation tools.
* **Potential Earnings**: Can reach 7 figures with specialization and reputation.

**Tips for Achieving 7 Figures:**

* **Specialize**: Focus on niche skills where demand outstrips supply.
* **Certifications**: Gain industry-recognized certifications (e.g., AWS Certified Solutions Architect, CISSP, etc.).
* **Networking**: Build a professional network to access high-value opportunities.
* **Continuous Learning**: Stay updated with emerging trends and technologies.

Which of these interests you the most? I can guide you on how to get started!

4o

**You said:**

can you tell me the geographical location of roman civilization?

**ChatGPT said:**

ChatGPT

The Roman civilization originated in the **Italian Peninsula** and expanded to become one of the largest empires in history, encompassing vast regions of Europe, the Mediterranean, and parts of Asia and Africa. Here's a detailed overview of its geographical extent:

**1. Core Region: The Italian Peninsula**

* **Rome**: The capital and the heart of the Roman civilization, situated on the banks of the Tiber River in central Italy.
* Other significant regions in Italy included:
  + **Latium** (where Rome is located)
  + **Campania**
  + **Etruria**

**2. Expansion Across Europe**

* **Western Europe**:
  + Modern-day **Spain** (Hispania)
  + **France** (Gaul)
  + **Portugal**
  + Parts of **Germany** (up to the Rhine River)
  + **Britain** (modern England and Wales, known as Britannia)
* **Central Europe**:
  + Parts of modern **Switzerland**, **Austria**, and **Hungary**.
* **Eastern Europe**:
  + Parts of the **Balkans**, including modern-day **Greece**, **Albania**, **Croatia**, and **Serbia**.

**3. Mediterranean and North Africa**

* **Mediterranean Islands**:
  + **Sicily**, **Sardinia**, **Corsica**, **Crete**, and other islands.
* **North Africa**:
  + Modern-day **Egypt**, **Libya**, **Tunisia** (where Carthage was located), **Algeria**, and parts of **Morocco**.

**4. Asia**

* **Asia Minor** (modern-day Turkey): Known as the province of Asia in the Roman Empire.
* **Levant**: Including parts of modern-day **Syria**, **Lebanon**, **Israel**, and **Jordan**.
* **Mesopotamia**: Briefly controlled parts of modern-day **Iraq** during its peak.

**5. Mediterranean Center**

* The Roman Empire surrounded the **Mediterranean Sea**, often called "Mare Nostrum" (Our Sea) by the Romans, as it was the central hub of their trade, culture, and military power.

**Timeline of Expansion:**

* **8th century BCE**: Rome's foundation in central Italy.
* **3rd–2nd century BCE**: Expansion across the western Mediterranean after defeating Carthage in the Punic Wars.
* **1st century BCE**: Julius Caesar's conquest of Gaul and expansion into Britain.
* **1st–2nd century CE**: Peak of the Roman Empire under emperors like Augustus and Trajan.

Would you like more details about any specific region or its cultural significance?

4o

**You said:**

I only want a small para

**ChatGPT said:**

ChatGPT

The Roman civilization originated on the **Italian Peninsula**, centered in the city of Rome, and expanded to become one of the largest empires in history. At its height, it spanned much of Europe, the Mediterranean, North Africa, and parts of Asia, including modern-day Italy, France, Spain, Greece, Egypt, Turkey, and Britain. The Mediterranean Sea served as the core of its trade, culture, and power, often referred to as "Mare Nostrum" (Our Sea) by the Romans.

3/3

4o

**You said:**

in easy language as if i read in class 3

**ChatGPT said:**

ChatGPT

The Roman civilization started in a city called **Rome**, in the middle of **Italy**. Over time, it grew very big and ruled over many places, including parts of **Europe**, **North Africa**, and **Asia**. It surrounded the **Mediterranean Sea**, which they called "Our Sea" because it was so important for their trade and travel.

4o

**You said:**

you didnt mention italy france spain greece and other stuffs you mentioned earlier

**ChatGPT said:**

ChatGPT

The Roman civilization started in **Rome**, a city in **Italy**, and grew to rule many places like **France**, **Spain**, **Greece**, **Egypt**, **Turkey**, and **Britain**. It was built around the **Mediterranean Sea**, which was very important for their trade and travel.

4o

**You said:**

just make it easy. dont have to exclude infos. The Roman civilization originated on the Italian Peninsula, centered in the city of Rome, and expanded to become one of the largest empires in history. At its height, it spanned much of Europe, the Mediterranean, North Africa, and parts of Asia, including modern-day Italy, France, Spain, Greece, Egypt, Turkey, and Britain. The Mediterranean Sea served as the core of its trade, culture, and power, often referred to as "Mare Nostrum" (Our Sea) by the Romans.

**ChatGPT said:**

ChatGPT

The Roman civilization started in **Rome**, a city in **Italy**. It grew to become one of the biggest empires in history. At its largest, it covered much of **Europe**, the **Mediterranean**, **North Africa**, and parts of **Asia**, including places like **Italy**, **France**, **Spain**, **Greece**, **Egypt**, **Turkey**, and **Britain**. The Romans called the **Mediterranean Sea** "Mare Nostrum" (Our Sea) because it was so important for their trade, culture, and power.

4o mini

**You said:**

I want to combine AI and robotics and ML...I know that you know everything....suppose you are a robotic engineer and you know everything....now teach me everything from beginning

**ChatGPT said:**

ChatGPT

Memory updated

To combine AI, robotics, and machine learning, you'll need to understand the fundamentals of each field and how they work together. Let’s break it down step by step, starting from the basics:

**1. Robotics Basics:**

Robotics involves creating machines that can perform tasks automatically. These machines can move, sense the environment, and make decisions.

**Key Concepts:**

* **Actuators:** Parts that enable movement (e.g., motors, wheels, robotic arms).
* **Sensors:** Devices that collect data about the environment (e.g., cameras, infrared sensors, accelerometers).
* **Controllers:** The brain of the robot, usually a computer or microcontroller (e.g., Arduino, Raspberry Pi).

**Learning Path:**

* Start with basic electronics and microcontrollers.
* Learn programming in C/C++ for controlling hardware (e.g., using Arduino or Raspberry Pi).
* Build simple robots, like a line-following robot, to get hands-on experience.

**2. Machine Learning (ML) Basics:**

Machine Learning enables systems to learn from data and improve without being explicitly programmed.

**Key Concepts:**

* **Supervised Learning:** The model is trained with labeled data (e.g., training a robot to recognize objects).
* **Unsupervised Learning:** The model finds patterns in unlabeled data (e.g., clustering similar objects).
* **Reinforcement Learning:** The model learns by interacting with the environment and getting rewards or penalties (important for robotics).

**Learning Path:**

* Start with Python, as it’s widely used in ML.
* Learn libraries like **NumPy**, **Pandas**, **Matplotlib** for data manipulation and visualization.
* Study ML algorithms such as linear regression, decision trees, and k-nearest neighbors.
* Practice using libraries like **scikit-learn** to implement these algorithms.

**3. AI in Robotics:**

AI in robotics allows robots to make decisions, plan their actions, and adapt to new situations.

**Key Concepts:**

* **Computer Vision:** Robots use cameras and sensors to “see” and interpret the world (e.g., object detection, facial recognition).
* **Path Planning:** AI algorithms help robots plan the best path to move through an environment, avoiding obstacles.
* **Natural Language Processing (NLP):** Robots can understand and respond to human speech (e.g., voice-controlled assistants).

**Learning Path:**

* Learn about image processing and computer vision using libraries like **OpenCV**.
* Explore AI techniques like **Deep Learning** and **Convolutional Neural Networks (CNNs)**, which are used in robotics for object recognition.
* Study reinforcement learning algorithms, which are essential for robots to make decisions in dynamic environments (e.g., Deep Q-Learning, Policy Gradient Methods).

**4. Combining AI, Robotics, and ML:**

To create robots that can learn and adapt using AI and ML, you'll need to integrate these concepts. For example, a robot can use ML to improve its navigation skills over time or use AI to make decisions based on its sensory input.

**Key Projects:**

* **Autonomous Vehicles:** Using sensors, computer vision, and machine learning to create self-driving robots (or cars).
* **Robotic Manipulation:** Teaching robots to handle objects using AI and ML techniques for grasping, picking, and moving things.
* **Humanoid Robots:** Combining all these technologies to create robots that can interact with humans and the environment like a person.

**5. Tools and Technologies:**

* **Programming Languages:** Python (for AI and ML), C/C++ (for low-level robotics control).
* **AI Libraries:** TensorFlow, PyTorch, Keras (for deep learning), OpenCV (for computer vision).
* **Robotics Frameworks:** ROS (Robot Operating System), VEX Robotics, Arduino, and Raspberry Pi.

**6. Hands-on Practice:**

* **Start with Simple Robots:** Use Arduino or Raspberry Pi to build small robots.
* **Explore Simulators:** Tools like **Gazebo** and **Webots** allow you to simulate robots and test AI algorithms before deploying them in real robots.
* **Projects:** Build projects like a robot that follows a path, a robot arm that picks and places objects, or a robot that navigates autonomously using AI and ML.

By mastering the basics of robotics, machine learning, and AI, and then working on projects that combine them, you’ll be well on your way to building intelligent robots. Start small, build your foundation, and gradually tackle more complex systems as you go. Would you like more resources or a roadmap for each area?

4o mini

**You said:**

teach me....I want detailed course from you

**ChatGPT said:**

ChatGPT

Let's begin your journey from the very basics and progressively build your knowledge in robotics, AI, and machine learning (ML). I'll guide you through a detailed step-by-step course, starting with fundamentals and moving toward more advanced topics.

**Phase 1: Understanding the Basics**

**1. Basic Programming and Electronics**

To control robots and implement AI/ML algorithms, you need to have a solid foundation in programming and electronics.

**a) Programming Fundamentals (Python)**

Start by learning **Python**, as it's widely used in both AI and robotics.

**Topics to cover:**

* Variables, Data Types, and Operators
* Control Structures (if/else, loops)
* Functions and Modules
* Lists, Tuples, Dictionaries, and Sets
* Object-Oriented Programming (Classes, Objects)
* Libraries: NumPy, Pandas for data manipulation

**Resources:**

* Python documentation: [Python Official Docs](https://docs.python.org/3/tutorial/)
* FreeCodeCamp Python Course: FreeCodeCamp

**b) Introduction to Electronics**

You'll need to know how basic electronic components work because robots interact with the physical world.

**Topics to cover:**

* **Resistors, Capacitors, Diodes, LEDs**
* **Breadboards and Circuits**
* **Power supply, Motors, and Sensors**
* **Microcontrollers (Arduino/Raspberry Pi)**

**Resources:**

* Arduino: Arduino Getting Started
* Raspberry Pi: Raspberry Pi Docs

**Phase 2: Introduction to Robotics**

**2. Building Simple Robots**

Once you're comfortable with programming and basic electronics, you can start building simple robots.

**a) Microcontroller Programming (Arduino / Raspberry Pi)**

**Topics to cover:**

* **Arduino Programming:** Learn how to write code that controls sensors and motors using Arduino.
* **Raspberry Pi Programming:** Learn how to interface with sensors and motors using Raspberry Pi and Python.

**Projects:**

* **Blink an LED:** Control an LED using code (basic, but crucial to understand control).
* **Servo Motor Control:** Use a motor to create simple movements like turning a wheel.

**Resources:**

* Arduino: Arduino Tutorials
* Raspberry Pi Robotics: Raspberry Pi Robotics

**b) Robot Sensors and Actuators**

**Topics to cover:**

* **Sensors**: Ultrasonic, infrared, cameras (basic image capture)
* **Actuators**: Motors, servos, and wheels (for movement)

**Projects:**

* **Line Following Robot**: A robot that uses sensors to follow a line.
* **Obstacle Avoidance Robot**: A robot that detects and avoids obstacles using ultrasonic sensors.

**Resources:**

* Arduino Sensors Guide

**Phase 3: Introduction to AI & Machine Learning (ML)**

**3. Basic AI Concepts**

Once you've built simple robots, the next step is to start integrating AI to make robots smarter.

**a) Machine Learning Fundamentals**

**Topics to cover:**

* **What is ML?**: Machine learning allows computers to learn from data and improve over time.
* **Types of ML**: Supervised learning, unsupervised learning, reinforcement learning.
* **Basic Algorithms**:
  + **Linear Regression**
  + **Logistic Regression**
  + **Decision Trees**
  + **K-Nearest Neighbors (KNN)**

**Resources:**

* **Coursera: Machine Learning by Andrew Ng** ([Coursera Link](https://www.coursera.org/learn/machine-learning))
* **Kaggle**: Kaggle Learn

**b) Computer Vision with AI**

Computer vision allows robots to see and interpret the world. It’s a critical part of making robots autonomous.

**Topics to cover:**

* **Image Processing Basics**
* **Edge Detection, Object Detection**
* **Deep Learning and CNNs (Convolutional Neural Networks)**

**Projects:**

* **Object Recognition**: Use a camera to recognize objects using pre-trained models like **YOLO** (You Only Look Once).
* **Face Recognition**: Implement a system that identifies faces from a live camera feed.

**Resources:**

* **OpenCV (Python)**: OpenCV Documentation
* **TensorFlow**: TensorFlow for Computer Vision

**Phase 4: Deep Dive into Robotics and AI Integration**

**4. Advanced Robotics Concepts**

**a) Path Planning and Navigation**

Robots need to figure out how to navigate around obstacles in their environment.

**Topics to cover:**

* **SLAM (Simultaneous Localization and Mapping)**: Technique used for autonomous navigation.
* **Pathfinding Algorithms**: A\* algorithm, Dijkstra’s algorithm.

**Projects:**

* **Robot Navigation**: Build a robot that can navigate through a maze or map.
* **SLAM Implementation**: Implement SLAM for indoor navigation.

**Resources:**

* [Robot Operating System (ROS)](https://www.ros.org/)
* SLAM Tutorial with ROS

**b) Reinforcement Learning in Robotics**

Reinforcement Learning (RL) is a type of machine learning where robots learn by interacting with the environment and getting feedback.

**Topics to cover:**

* **Q-learning**: A simple RL algorithm.
* **Deep Q-Networks (DQN)**: A method where deep learning is combined with Q-learning.
* **Policy Gradient Methods**: Advanced RL for continuous action spaces.

**Projects:**

* **RL for Robot Arm**: Teach a robot arm to pick and place objects.
* **Autonomous Navigation**: Use RL to make a robot learn how to navigate a complex environment.

**Resources:**

* **OpenAI Gym**: [OpenAI Gym](https://gym.openai.com/)
* **RL Course by David Silver**: [David Silver’s RL Course](http://www.davidsilver.uk/)

**Phase 5: Final Projects & Research**

Now, you’ll combine all your skills into building sophisticated robots that use AI and ML to solve complex tasks.

**Projects:**

* **Autonomous Car**: Build a car that can drive itself using a combination of sensors, computer vision, and reinforcement learning.
* **Robotic Arm**: Teach a robot arm to perform complex tasks like assembling products or performing surgery.
* **Humanoid Robot**: Create a robot that can interact with humans using speech, vision, and physical actions.

**Resources:**

* **Google AI**: [Google AI Resources](https://ai.google/)
* **Robotics with Python**: [Python Robotics Projects](https://github.com/Adafruit/Adafruit_Learning_System_Guides/tree/master/Python_Robotics)

**Final Thoughts:**

By following this structured approach, you’ll go from building simple robots to implementing AI and machine learning to create autonomous systems. Robotics is a vast field, so as you gain more experience, you’ll discover specialized areas that you might want to dive deeper into, like robotic vision, manipulation, or autonomous vehicles.

Would you like specific resources or help with one of these phases?

4o mini

**You said:**

I need more free full fledged resources

**ChatGPT said:**

ChatGPT

Here’s a list of **free, full-fledged resources** that will help you learn robotics, AI, and machine learning from scratch, covering both theory and practical implementation.

**1. Basic Programming (Python)**

Python is the most widely used programming language in both AI and robotics.

* [**Automate the Boring Stuff with Python**](https://automatetheboringstuff.com/) – Free book to learn Python, covering automation tasks and basic programming concepts.
* [**Python.org**](https://docs.python.org/3/tutorial/) – The official Python documentation, which includes a complete tutorial for beginners.
* **FreeCodeCamp Python** – FreeCodeCamp offers a comprehensive Python course from beginner to advanced level.

**2. Introduction to Electronics and Robotics**

* **Adafruit Learn System** – A huge collection of tutorials on building electronics and robotics projects using Raspberry Pi, Arduino, and other components.
* **Arduino Official Tutorials** – Free tutorials on programming and building projects with Arduino.
* **Raspberry Pi Documentation** – Documentation and tutorials for Raspberry Pi, ideal for beginners in robotics and electronics.

**3. Machine Learning**

* **Google's Machine Learning Crash Course** – A free course covering the basics of machine learning with TensorFlow, featuring video lectures and practical exercises.
* **Kaggle Learn** – Free, hands-on, and beginner-friendly courses on machine learning, deep learning, and AI techniques.
* [**Coursera – Machine Learning by Andrew Ng**](https://www.coursera.org/learn/machine-learning) – Free to audit, one of the most popular and comprehensive ML courses.
* [**DeepLizard YouTube Channel**](https://www.youtube.com/c/DeepLizard) – DeepLizard provides a free and well-explained series of videos on machine learning, neural networks, and reinforcement learning.

**4. AI and Machine Learning Libraries**

* **TensorFlow for Beginners** – Free tutorials and documentation from Google on how to use TensorFlow for machine learning.
* **PyTorch Documentation** – PyTorch is another popular deep learning library; their tutorials cover everything from beginner concepts to advanced deep learning models.
* **Scikit-learn Documentation** – Scikit-learn is a popular Python library for traditional machine learning. It has excellent documentation with easy-to-understand examples.

**5. Computer Vision (AI for Robots)**

* **OpenCV Documentation** – OpenCV is one of the most widely used libraries for computer vision and image processing. The documentation offers both beginner and advanced tutorials.
* [**PyImageSearch**](https://pyimagesearch.com/) – A blog with free tutorials on computer vision, deep learning, and OpenCV.
* [**Coursera – Computer Vision Basics**](https://www.coursera.org/learn/computervision-basics) – Free course on computer vision basics and applications (you can audit it for free).

**6. Reinforcement Learning and Robotics**

* [**OpenAI Gym**](https://gym.openai.com/) – A toolkit for developing and comparing reinforcement learning algorithms. It includes a variety of environments for testing your RL models.
* [**David Silver’s Reinforcement Learning Course**](http://www.davidsilver.uk/) – A well-regarded free course on reinforcement learning. David Silver’s videos are great for building a solid understanding of RL.
* **MIT OpenCourseWare – Introduction to Robotics** – Free lecture notes and video tutorials on robotics from MIT.

**7. Advanced Robotics and AI Integration**

* [**Robot Operating System (ROS)**](https://www.ros.org/) – ROS is the standard framework for building robotics software. Their site provides free tutorials on how to use ROS for robotic programming.
* **ROS Tutorials** – Free tutorials to learn ROS and develop robots that can perform real-world tasks.
* [**Coursera – Robotics Specialization**](https://www.coursera.org/specializations/robotics) – This Coursera series is a comprehensive free course on robotics, covering everything from kinematics to machine learning for robots (you can audit the course for free).

**8. Robotics Simulators and Virtual Robotics**

* [**Webots**](https://www.cyberbotics.com/) – A free robotics simulator to help you simulate robot behaviors and test your code without needing physical hardware.
* **VEX Robotics** – A robotics platform that offers free software to program robots with educational resources for beginners.
* [**Gazebo Simulator**](http://gazebosim.org/) – A powerful and free robot simulation tool used in conjunction with ROS for building complex robots.

**9. Robotics Projects and GitHub Repositories**

* [**GitHub Robotics Projects**](https://github.com/topics/robotics) – Browse through open-source projects on GitHub for robotics. You can fork and contribute to ongoing projects.
* [**Makeblock**](https://www.makeblock.com/) – Offers free open-source robotics kits and guides, ideal for beginners who want to build projects using robotics components.

**10. Online Communities and Forums**

* [**Stack Overflow**](https://stackoverflow.com/) – A great place to ask technical questions related to robotics, AI, and programming.
* [**Robot Stack Exchange**](https://robotics.stackexchange.com/) – A community specifically for robotics, where you can ask and answer questions related to robotics.
* [**Kaggle**](https://www.kaggle.com/) – Join Kaggle to practice machine learning and computer vision problems and participate in competitions.

**11. Research Papers and Advanced Learning**

For more advanced, in-depth understanding, you can read papers and research articles related to AI and robotics:

* [**ArXiv**](https://arxiv.org/) – A collection of research papers across many fields of science, including AI, machine learning, and robotics. Many of these papers are available for free.
* [**Google Scholar**](https://scholar.google.com/) – Search for academic papers and research articles on topics like AI, ML, and robotics.

These resources should help you cover every aspect of robotics, AI, and ML, from the very basics to advanced topics. As you learn and work on projects, you will gradually integrate AI and ML with robotics. Feel free to explore these resources at your own pace, and don't hesitate to ask if you need help on specific topics!