

Chapter 4: System Troubleshooting

LONG Question from the Chapter “System Troubleshooting”:

EXERCISE LONG Question with Answers

Q1. Discuss the importance of troubleshooting in maintaining the smooth operation of systems, especially computing systems.

Troubleshooting is the process of identifying, analyzing, and solving problems in a system. It is highly important for maintaining smooth and efficient computer operations.

1. Ensures smooth working:

Without troubleshooting, even minor issues like a loose cable, faulty driver, or virus can stop the system. Fixing these ensures continuous performance.

2. Saves time and cost:

Quick identification and fixing reduces repair expenses and prevents wasting time on unnecessary replacements.

3. Prevents downtime:

System failures reduce productivity. Troubleshooting minimizes downtime and allows work to continue without major interruptions.

4. Improves reliability:

When issues are solved quickly, users trust the system for important work. This reliability is necessary for both offices and schools.

5. Examples:

- Fixing a computer that won't boot due to loose RAM.
- Repairing a printer error by updating drivers.
- Removing a virus that slows down the system.

Conclusion: Troubleshooting keeps systems efficient, prevents loss of work, and maintains user trust in technology.

Q2. Explain the systematic process of troubleshooting. Describe each step in detail.

Troubleshooting is a structured process followed step by step:

1. Identify the problem:

Observe and confirm the issue. Example: The computer does not turn on.

2. Establish a theory of probable cause:

Think of possible reasons, faulty cable, dead battery, or power supply problem.

3. Test the theory:

Check if the suspected cause is true. Example: Replacing the cable to see if the computer starts.

4. Establish a plan of action and implement the solution:

Apply the confirmed fix. For example, replacing a faulty power supply.

5. Verify full functionality:

Check if the system works properly after repair. Ensure no new problems appear.

6. Document findings, actions, and outcomes:

Write down the problem, solution, and result so it can help in the future.

Conclusion: This systematic approach ensures accuracy, avoids guesswork, and helps in solving problems efficiently.

Q3. Using a case study where a printer is not printing, explain how you would identify the problem and establish a theory of probable cause.**Case Study: Printer not printing****1. Identify the problem:**

The user sends a print command but the printer produces no output.

2. Gather information:

Check if the printer is powered on, connected properly, and has paper.

3. Establish theories of cause:

- No paper in the tray
- USB cable disconnected
- Printer driver missing or corrupted
- Empty ink cartridge

4. Select most probable cause:

Suppose the driver is missing or corrupted.

5. Test the theory:

Reinstall the driver and check. If printing works, the theory is correct.

Conclusion: By carefully identifying and testing possible causes, we can solve printer issues without wasting resources.

Q4. Discuss the importance of documenting findings, actions, and outcomes during the troubleshooting process.

1. Record keeping for future use:

Documentation helps solve similar issues quickly later.

2. Saves time and effort:

Technicians don't need to repeat the full process; they can refer to past records.

3. Knowledge sharing:

Other team members can understand what was done, even if they were not present.

4. Professionalism:

Maintains an organized and systematic record, which increases trust.

5. Example:

If a laptop does not start because of a faulty RAM slot, documenting this can save hours of testing when the same problem occurs again.

Conclusion: Documentation is not optional but a vital step in professional troubleshooting.

Q5. Analyze the various ways troubleshooting is vital in computing systems, particularly in preventing downtime, ensuring data integrity, and improving security.

1. Preventing downtime:

Quick detection of faults like power supply failure prevents loss of work hours. Example: Fixing a boot problem before office work begins.

2. Ensuring data integrity:

Data must remain safe. Troubleshooting disk issues or file system errors protects valuable information. Example: Recovering corrupted files by repairing storage.

3. Improving security:

Software problems like malware can harm data. Troubleshooting detects and removes these threats, keeping systems secure. Example: Running antivirus to remove infections.

4. Improving productivity:

When problems are solved quickly, users can focus on work instead of waiting for repairs.

5. Reliability and trust:

Stable systems build confidence among users and prevent frustration.

Conclusion: Troubleshooting is not just repair; it is protection of data, security, and continuous workflow.

Q6. Describe basic software-related troubleshooting strategies, including handling application freezing and unresponsive peripherals.**1. Application freezing:**

- End the task using Task Manager.
- Restart the application to continue work.

2. Unresponsive peripherals:

- Reinstall or update device drivers.
- Reconnect the device properly.

3. Checking system updates:

Update the operating system to fix compatibility issues.

4. Restarting the system:

Many temporary software errors are cleared after a restart.

5. Example:

If MS Word freezes, ending the task in Task Manager and restarting it helps resume work.

Conclusion: These strategies ensure smooth working without requiring advanced repair methods.

Q7. Explain how to recognize hardware failures, particularly focusing on RAM and hard drive issues.**1. RAM issues:**

- Frequent restarts
- System crashes while running programs
- Blue screen errors

2. Hard drive issues:

- System not booting
- Missing or corrupted files
- Very slow system performance

3. Importance of recognition:

Early detection saves data and prevents complete system failure.

4. Example:

A laptop restarting repeatedly may indicate faulty RAM. A hard disk making clicking noises may be failing.

Conclusion: Recognizing these symptoms helps in quick replacement and system recovery.

Q8. Elaborate on the importance of maintaining software and addressing security threats.**1. Software maintenance:**

Regular updates ensure smooth working and prevent errors.

2. Preventing crashes:

Updated software reduces freezing and program errors.

3. Addressing threats:

Viruses and malware damage data. Antivirus software protects against them.

4. Security patches:

Operating system updates close security holes and make systems safe.

5. Example:

Updating antivirus prevents infections from new viruses.

Conclusion: Maintenance and security measures are necessary for safe and reliable systems.

Q9. Describe common methods for identifying and removing malware infections and applying operating system updates for security.**1. Identifying malware:**

Look for slow speed, pop-ups, or corrupted files.

2. Removing malware:

Run antivirus software to scan and clean infections.

3. Updating OS:

Install updates to fix weak points in the system.

4. Verification:

Check that system works smoothly after cleaning and updating.

5. Example:

Removing a virus that deletes files and installing updates to prevent further attacks.

Conclusion: These steps ensure safety, protect data, and strengthen system security.

Q10. Describe the different data backup methods, including using external storage devices and cloud solutions.**1. External storage devices:**

USB drives, DVDs, and external hard disks are used to copy and store important files.

2. Cloud solutions:

Files are stored online and can be accessed anytime with internet.

3. Importance:

Backup protects against hardware failure, corruption, or accidental deletion.

4. Examples:

- Saving school projects in a USB.
- Storing photos in Google Drive.

5. Conclusion:

Data backup ensures safety and availability of information, even if the main system is damaged.

Additional LONG Question with Answers

1. Explain the difference between preventive and corrective maintenance. Discuss their importance with examples.
 2. Describe the common problems of input, output, and storage devices. How can each type be troubleshooted?
 3. What are the major causes of software-related problems in computers? How can these problems be solved effectively?
 4. Discuss the role of troubleshooting tools such as multimeter, antivirus, and Task Manager in solving computer problems.
 5. Explain the safety precautions that should be followed during troubleshooting to protect both the user and the computer system.
 6. Describe step by step how you would troubleshoot a computer that is not starting at all.
 7. Discuss how viruses and malware affect system performance and explain the troubleshooting steps to remove them.
 8. Explain the importance of data backup in system troubleshooting. Discuss different backup strategies with their advantages.
 9. Describe the impact of not documenting troubleshooting activities. Why is proper documentation necessary for long-term maintenance?
 10. Explain how troubleshooting contributes to system security and reliability. Give real-life examples.
-