 Which statement correctly includes header.php, halting script execution on failure?

a) include 'header.php';

b) **require 'header.php'; (✓)**

c) require_once();

d) import 'header.php';

 Which keyword *does not* stop script execution when the file is missing?

a) require

b) **include (✓)**

c) require_once

d) All of the above

 What happens if require_once 'config.php'; is executed after config.php was already required?

a) Includes again

b) **Ignored (✓)**

c) Fatal error

d) Warning only

 Which construct returns true on success and false on failure while continuing execution?

a) **include (✓)**

b) require

c) require_once

d) None

 include_path is configured by which php.ini directive?

a) open_basedir

b) auto_prepend_file

c) **include_path (✓)**

d) extension_dir

 If require cannot find a file, PHP will:

a) Notice and continue

b) Warning and continue

c) **Fatal-error and exit (✓)**

d) Silently ignore

 Which wrapper allows file inclusion over HTTP when allow_url_include is enabled?

a) file://

b) ftp://

c) **http:// (✓)**

d) php://input

 How can you suppress include warnings?

a) silent include 'x.php';

b) !include 'x.php';
c) **@include 'x.php'; (✓)**
d) Impossible

 Which superglobal is inherently modified by an include?
a) $GLOBALS
b) $_SERVER
c) $_POST
d) **None; include doesn't automatically alter them (✓)**

 Turning display_errors = Off primarily affects:
a) **Whether include/require messages appear in the browser (✓)**
b) Execution flow of require
c) include-path order
d) File encoding

 Given include_path=".:/usr/share/php", search order is:
a) /usr/share/php then .
b) **. (current dir) then /usr/share/php (✓)**
c) Alphabetical
d) Random

 Which function appends to the current include path at runtime?
a) ini_alter()
b) ini_set()
c) **set_include_path() (✓)**
d) alter_include_path()

 Which magic constant reveals the absolute filename of the current script?
a) __DIR__
b) **__FILE__ (✓)**
c) __LINE__
d) __PATH__

 require_once can speed things up mainly by:
a) **Avoiding re-reading already-included files (✓)**
b) Forcing opcode cache
c) Shrinking bytecode
d) Disabling GC

 A path segment beginning with / inside include_path means:
a) **Absolute path (search stops here) (✓)**
b) Relative path
c) URL stream wrapper
d) NFS mount

 Two scripts in different dirs both run `require_once 'lib.php';`. `lib.php` executes:
a) 0 times
b) **1 time (✓)**
c) 2 times
d) Depends on OS

 Functions declared in an included file are:
a) Unavailable
b) Tied to a special namespace
c) **Globally available within the request (✓)**
d) Scope-limited to the include

 Variables declared before `include` inside a function are:
a) **Local to that function (✓)**
b) Automatically global
c) Passed by ref
d) Superglobals

 Keyword to modify a parent-scope variable from inside an include:
a) **`global` (✓)**
b) `static`
c) `extern`
d) `shared`

 Best magic constant for portable, file-relative include paths:
a) `__LINE__`
b) `__FUNCTION__`
c) `__FILE__`
d) **`__DIR__` (✓)**

 Including a directory results in:
a) Directory listing
b) **Warning + `false` (✓)**
c) Exception
d) Autoload

 PHP 8: including a bad URL with wrappers on triggers:
a) TypeError
b) ErrorException
c) **Warning (✓)**
d) Shutdown handler

 Automatic class-file loading uses:
a) **`spl_autoload_register()` (✓)**
b) `autoload_classes()`

c) include_path
d) PSR-0 only

 Composer PSR-4 chiefly eliminates:
a) Per-env include-path hacks
b) **Manual `require` per class (✓)**
c) Need for opcache
d) Use of closures

 Sensitive config files should live outside web-root because they might:
a) Cache forever
b) **Be downloaded as plain text if mis-served (✓)**
c) Upsize DNS
d) Disable buffering

 `include_once` decides duplicates by comparing:
a) file hashes
b) **Resolved absolute paths (✓)**
c) Original relative paths
d) Timestamps

 Inside a namespace, `require 'foo.php';` executes `foo.php`:
a) In that namespace
b) **In global namespace (unless `foo.php` declares one) (✓)**
c) As a trait
d) Only via autoload

 Modifying `include_path` with `ini_set()` affects:
a) **Subsequent include/require in this request (✓)**
b) All requests
c) CLI only
d) Ignored since PHP 8

 Function listing all included/required files so far:
a) LIST_INCLUDES
b) **`get_included_files()` (✓)**
c) __INCLUDED__
d) debug_backtrace()

 Guarantee a file is included exactly once *and* fatal on failure:
a) require 'file.php';
b) **`require_once 'file.php';` (✓)**
c) include 'file.php';
d) include_once 'file.php';

1. Which PHP function is commonly used to change the permissions of an existing file or directory on the server?

a) chmod()
b) chown()
c) fopen()
d) umask()

Answer: a) chmod()

---

2. Given the statement `chmod("report.txt", 0644);`, what permission set is applied to report.txt on a typical UNIX-like system?

a) Owner: read & write, Group: read, Others: read
b) Owner: read, write & execute, Group: read & execute, Others: read & execute
c) Owner: read & write, Group: write, Others: write
d) Owner: read, Group: read & write, Others: execute

Answer: a) Owner: read & write, Group: read, Others: read

---

3. In a default Apache setup running as user www-data, which of the following permission octal values would normally prevent PHP from writing to `upload/` while still allowing uploads if ownership is corrected?

a) 0777
b) 0755
c) 0644
d) 0700

Answer: b) 0755
*(If ownership is **www-data:www-data**, 0755 grants write only to the owner, satisfying security best-practices.)*

---

4. What does the leading zero in the octal literal `0750` signify in a `chmod()` call?

a) It marks the value as hexadecimal.
b) It is required for backward compatibility with PHP 4.
c) It tells PHP to interpret the number as octal (base 8).
d) It has no effect; it can be omitted.

Answer: c) It tells PHP to interpret the number as octal (base 8).

---

5. Which umask setting should you apply before creating a log file with `fopen()` so that the resulting file has permissions rw-r----- (0640)?

a) umask(000)
b) umask(022)
c) umask(027)
d) umask(002)

Answer: c) umask(027)

(Default creation 0666 minus 027 ➔ 0640.)

---

6. Using `fileperms()` returns 16877 for a directory. What symbolic permission string does this represent?

a) drwxr-xr-x
b) drwxrwxrwx
c) -rw-r--r--
d) drwx------

Answer: a) drwxr-xr-x
(16877 decimal = 040755 octal; leading 04 indicates directory.)

---

7. Which of these is NOT a valid reason for getting a "Permission denied" error in PHP when writing to a file?

a) The directory is mounted read-only by the OS.
b) SELinux or AppArmor policies restrict the script.
c) The PHP open_basedir directive blocks the path.
d) The script forgot to call clearstatcache() before writing.

Answer: d) The script forgot to call clearstatcache() before writing.
(clearstatcache() affects cached stat results, not permissions.)

---

8. On shared hosting, why is setting a file to 0777 generally discouraged even if it "fixes" permission errors?

a) It disables PHP's memory limit.
b) It grants write access to every user on the server, posing security risks.
c) It causes PHP to treat the file as binary.
d) It slows down file I/O operations.

Answer: b) It grants write access to every user on the server, posing security risks.

---

9. To ensure that newly uploaded images are not executable on an Nginx/PHP-FPM stack, which permission set is recommended?

a) 0644
b) 0755
c) 0700
d) 0775

Answer: a) 0644
(No execute bit for anyone, read for web server to serve the image.)

---

10. When you run PHP via CLI as your normal user, `touch("notes.txt");` creates the file with `rw-rw-r--`. Which `umask` is most likely in effect?

a) 0002
b) 0022
c) 0277
d) 0777

Answer: a) 0002

1. Which PHP function is used to open a file?

a) open_file()

b) file_open()

c) fopen()

d) open()

Answer: c) fopen()

2. What is the purpose of the second argument in the fopen() function?

a) The file size.

b) The file path.

c) The file access mode.

d) The file permissions.

Answer: c) The file access mode.

3. Which file access mode in fopen() opens a file for writing only, placing the file pointer at the end of the file?

a) w

b) r+

c) a

d) x

Answer: c) a

4. Which file access mode in fopen() opens a file for reading and writing; creates a new file if it doesn't exist, otherwise overwrites the existing file?

a) r+

b) w+

c) a+

d) x+

Answer: b) w+

5. What function is used to close an open file in PHP?

a) close_file()

b) file_close()

c) fclose()

d) close()

Answer: c) fclose()

6. Which PHP function reads the entire content of a file into a string?

a) readfile()

b) fread()

c) file_get_contents()

d) get_file()

Answer: c) file_get_contents()

7. What function reads a single line from a file pointer?

a) fgets()

b) fgetline()

c) freadln()

d) readline()

Answer: a) fgets()

8. Which PHP function reads a specified number of bytes from a file pointer?

a) fread()

b) freadbytes()

c) fgetbytes()

d) readbytes()

Answer: a) fread()

9. What function writes content to an open file?

a) fwrite()

b) file_write()

c) write_file()

d) fput()

Answer: a) fwrite()

10. Which PHP function writes the entire content of a string to a file?

a) fwrite_all()

b) file_put_contents()

c) put_file()

d) string_to_file()

Answer: b) file_put_contents()

11. What function checks if a file or directory exists?

a) file_exists()

b) is_exist()

c) exist_file()

d) check_file()

Answer: a) file_exists()

12. Which function is used to delete a file?

a) delete_file()

b) remove_file()

c) unlink()

d) erase()

Answer: c) unlink()

13. What function renames a file or directory?

a) change_name()

b) rename_file()

c) move_file()

d) rename()

Answer: d) rename()

14. Which function copies a file?

a) file_copy()

b) copy_file()

c) duplicate_file()

d) copy()

Answer: d) copy()

15. What function returns the size of a file in bytes?

a) filesize()

b) size_file()

c) get_file_size()

d) file_length()

Answer: a) filesize()

16. Which function checks if a given path is a directory?

a) is_dir()

b) is_directory()

c) check_dir()

d) dir_exists()

Answer: a) is_dir()

17. What function creates a directory?

a) make_dir()

b) create_dir()

c) mkdir()

d) new_dir()

Answer: c) mkdir()

18. Which function removes a directory?

a) remove_dir()

b) delete_dir()

c) rmdir()

d) erase_dir()

Answer: c) rmdir()

19. What function returns an array of files and directories within a specified path?

a) get_files()

b) list_files()

c) scandir()

d) read_dir()

Answer: c) scandir()

20. Which function changes the permissions of a file or directory?

a) change_perms()

b) set_permissions()

c) chmod()

d) file_perms()

Answer: c) chmod()

21. What does the acronym CSV stand for in the context of file input/output?

a) Comma Separated Value

b) Character Separated Variable

c) Common Standard Value

d) Compiled System Variable

Answer: a) Comma Separated Value

22. Which PHP function can be used to parse a CSV file?

a) readcsv()

b) parse_csv()

c) fgetcsv()

d) getcsv()

Answer: c) fgetcsv()

23. When reading a CSV file with fgetcsv(), what is the default delimiter?

a) Semicolon (;)

b) Tab (\t)

c) Comma (,)

d) Space ()

Answer: c) Comma (,)

24. What function is often used to output data in a structured format that can be easily read by other applications or stored in files?

a) print_r()

b) var_dump()

c) json_encode()

d) serialize()

Answer: c) json_encode()

25. Which function is used to decode a JSON string back into a PHP variable?

a) json_decode()

b) decode_json()

c) unjson()

d) parse_json()

Answer: a) json_decode()

26. What is the purpose of the flock() function in PHP?

a) To format a file.

b) To lock or release a file.

c) To find a file.

d) To flush the output buffer to a file.

Answer: b) To lock or release a file.

27. Which lock type in flock() acquires an exclusive lock for writing?

a) LOCK_SH

b) LOCK_EX

c) LOCK_UN

d) LOCK_NB

Answer: b) LOCK_EX

28. What does LOCK_UN flag in flock() do?

a) Locks the file shared.

b) Locks the file exclusively.

c) Releases a lock.

d) Makes the lock non-blocking.

Answer: c) Releases a lock.

29. What function is used to get information about a file, such as its access time, modification time, etc.?

a) fileinfo()

b) file_info()

c) stat()

d) get_file_info()

Answer: c) stat()

30. Which of the following is NOT a valid file access mode for fopen()?

a) r

b) w

c) z

d) a

Answer: c) z

- Which command is commonly used in the command line to create a new directory?

a) `cd`

b) `rm`

c) `mkdir`

d) `ls`

`Answer:` c) `mkdir`

- What does `mkdir` stand for?

a) Move Directory

b) Make Directory

c) Modify Directory

d) Manage Directory

`Answer:` b) Make Directory

- To create a directory named "reports" in the current working directory, which command would you use?

a) `cd reports`

b) `rm reports`

c) `mkdir reports`

d) `ls reports`

`Answer:` c) `mkdir reports`

- Which option for the `mkdir` command allows you to create parent directories if they don't exist?

a) `-r`

b) `-f`

c) `-p`

d) `-v`

`Answer:` c) `-p`

- If you want to create a directory named "data" inside a directory named "projects" (and "projects" might not exist), which command would you use?

a) `mkdir data/projects`

b) `mkdir -p data/projects`

c) `mkdir projects/data`

d) `mkdir -p projects/data`

`Answer:` d) `mkdir -p projects/data`

- Which command is used to remove a directory?

a) `rd`

b) `del`

c) `rmdir`

d) `erase`

`Answer:` c) `rmdir`

- What is the primary difference between `rmdir` and `rm -r` when deleting directories?

a) `rmdir` can only delete empty directories.

b) `rm -r` can only delete empty directories.

c) There is no difference; they do the same thing.

d) `rmdir` is used for files, and `rm -r` is used for directories.

Answer: a) `rmdir` can only delete empty directories.

- To delete an empty directory named "temp", which command would you use?

a) `rm temp`

b) `rm -r temp`

c) `rmdir temp`

d) `del temp`

Answer: c) `rmdir temp`

- To recursively delete a directory named "backup" and all its contents (files and subdirectories), which command would you use?

a) `rmdir backup`

b) `rm backup`

c) `rm -r backup`

d) `del /f /q backup`

Answer: c) `rm -r backup`

- What does the `-r` option in the `rm` command stand for?

a) Remove forcefully

b) Remove recursively

c) Remove silently

d) Remove only files

Answer: b) Remove recursively

- Which command is used to change the current working directory?

a) `md`

b) `rd`

c) `cd`

d) `pwd`

`Answer:` c) `cd`

- What does `cd` stand for?

a) Change Directory

b) Current Directory

c) Create Directory

d) Command Directory

`Answer:` a) Change Directory

- To move from your current directory to a subdirectory named "images", which command would you use?

a) `cd ..`

b) `cd /images`

c) `cd images`

d) `move images`

`Answer:` c) `cd images`

- To move one level up in the directory hierarchy, which command would you use?

a) `cd .`

b) `cd /`

c) `cd ..`

d) `cd ../..`

`Answer:` c) `cd ..`

- What does the single dot (`.`) represent in directory navigation?

a) The root directory

b) The parent directory

c) The current directory

d) The home directory

`Answer:` c) The current directory

- What do the double dots (`..`) represent in directory navigation?

a) The current directory

b) The root directory

c) The parent directory

d) The home directory

`Answer:` c) The parent directory

- To go directly to your home directory from any location in the file system, you can often use the command:

a) `cd /home`

b) `cd ~`

c) `cd .`

d) `cd root`

`Answer:` b) `cd ~`

- What does the tilde symbol (~) typically represent in command-line environments?

a) The current directory

b) The root directory

c) The user's home directory

d) The parent directory

`Answer:` c) The user's home directory

- Which command displays the path of the current working directory?

a) `ls`

b) `pwd`

c) `cd`

d) `dir`

`Answer:` b) `pwd`

- What does `pwd` stand for?

a) Print Working Directory

b) Present Working Directory

c) Previous Working Directory

d) Path of Working Directory

`Answer:` a) Print Working Directory

- If your current directory is `/home/user/documents` and you want to change to `/home/user/downloads`, which command could you use?

a) `cd documents/downloads`

b) `cd ../downloads`

c) `cd /home/user/downloads`

d) Both b and c

`Answer:` d) Both b and c

- Which of the following commands would create a directory named "new_folder" and then move you into it?

a) `cd new_folder`

b) `mkdir new_folder && cd new_folder`

c) `mkdir -p new_folder`

d) `create new_folder; change new_folder`

`Answer:` b) `mkdir new_folder && cd new_folder`

- Which command would you use to delete a directory named "old_files" even if it contains files and subdirectories, without prompting for confirmation?

a) `rmdir old_files`

b) `rm -i -r old_files`

c) `rm -f -r old_files`

d) `del old_files /q`

`Answer:` c) `rm -f -r old_files`

- What is the purpose of the `-v` (verbose) option in the `mkdir` command?

a) To create parent directories.

b) To remove existing directories before creating.

c) To display a message for each created directory.

d) To suppress error messages.

Answer: c) To display a message for each created directory.

- Suppose you are in the `/usr/local` directory. Which command would take you to the root directory (`/`)?

a) `cd .`

b) `cd ..`

c) `cd ../..`

d) `cd /`

`Answer:` d) `cd /`

- You are in the directory `/home/user/data/images`. Which command would take you directly to `/home/user`?

a) `cd ..`

b) `cd ../..`

c) `cd /home/user`

d) `cd ../../..`

`Answer:` b) `cd ../..`

- Which of the following is NOT a standard command for working with directories in most command-line interfaces?

a) `mkdir`

b) `rmdir`

c) `cd`

d) `movedir`

`Answer:` d) `movedir`

- What happens if you try to use `rmdir` on a directory that is not empty?

a) It will delete the directory and its contents.

b) It will display an error message.

c) It will empty the directory and then delete it.

d) It will move the contents to the parent directory and then delete the original directory.

Answer: b) It will display an error message.

- Which command can be used to create multiple directories at once?

a) `mkdir dir1, dir2, dir3`

b) `mkdir dir1 & mkdir dir2 & mkdir dir3`

c) `mkdir dir1 dir2 dir3`

d) `create directories dir1 dir2 dir3`

`Answer:` c) `mkdir dir1 dir2 dir3`

- You want to create a directory named "logs" and then immediately navigate into it. Which of the following sequences of commands would achieve this?

a) `cd logs; mkdir logs`

b) `mkdir logs && cd logs`

c) `mkdir logs || cd logs`

d) `cd logs && mkdir logs`

`Answer:` b) `mkdir logs && cd logs`

- Which HTTP method is essential for submitting file uploads via HTML forms?

a) GET

b) POST

c) PUT

d) DELETE

Answer: b) POST

- What `enctype` attribute is mandatory in the `<form>` tag for file uploads?

a) `application/x-www-form-urlencoded`

b) `text/plain`

c) `multipart/form-data`

d) `application/octet-stream`

Answer: c) `multipart/form-data`

- In PHP, where is the information about uploaded files primarily stored?

a) `$_GET`

b) `$_POST`

c) `$_SESSION`

d) `$_FILES`

Answer: d) `$_FILES`

- To access the original name of an uploaded file (from an input named `userfile`), you would use:

a) `$_FILES['userfile']['name']`

b) `$_POST['userfile_name']`

c) `$_GET['userfile']`

d) `$_FILE['name']['userfile']`

Answer: a) `$_FILES['userfile']['name']`

- Which key in the `$_FILES` array holds the temporary location of the uploaded file on the server?

a) `name`

b) `size`

c) `type`

d) `tmp_name`

Answer: d) `tmp_name`

- Which PHP function is used to move an uploaded file from its temporary location to a permanent directory?

a) `copy()`

b) `rename()`

c) `move_uploaded_file()`

d) `file_put_contents()`

Answer: c) `move_uploaded_file()`

- What is the return value of `move_uploaded_file()` upon successfully moving the file?

a) The new file path

b) The file size

c) `true`

d) `false`

Answer: c) `true`

- A crucial security practice when handling file uploads is to verify the file's:

a) Temporary name

b) Original name

c) Extension and MIME type on the server-side

d) Size on the client-side

Answer: c) Extension and MIME type on the server-side

- Which PHP configuration directives often limit the maximum size of uploaded files?

a) `max_file_size` and `post_size_limit`

b) `upload_max_size` and `post_max_upload`

c) `upload_max_filesize` and `post_max_size`

d) `file_upload_limit` and `max_post_size`

Answer: c) `upload_max_filesize` and `post_max_size`

- Where are uploaded files initially stored on the server before being processed by PHP?

a) The webroot

b) A temporary directory specified in `php.ini`

c) The script's directory

d) A database

Answer: b) A temporary directory specified in `php.ini`

- What is a common method to prevent overwriting files with the same name during uploads?

a) Forcing users to provide unique names

b) Automatically deleting older files

c) Renaming uploaded files using timestamps or unique hashes

d) Limiting the number of uploads

Answer: c) Renaming uploaded files using timestamps or unique hashes

- What does `$_FILES['userfile']['error']` having a value of `0` indicate?

a) A file size limit was exceeded.

b) The file was only partially uploaded.

c) The file was uploaded successfully.

d) No file was uploaded.

Answer: c) The file was uploaded successfully.

- What does an `$_FILES['userfile']['error']` value of `UPLOAD_ERR_NO_FILE` signify?

a) The uploaded file is too large.

b) No file was selected for upload.

c) The temporary directory is missing.

d) A file upload was interrupted.

Answer: b) No file was selected for upload.

- When handling multiple file uploads from an input with `multiple`, how do you typically access the individual file information in `$_FILES` (for an input named `files[]`)?

a) Directly as `$_FILES['files'][0]`, `$_FILES['files'][1]`, etc.

b) As separate variables like `$file1`, `$file2`.

c) Through a loop iterating over `$_FILES['files']['name']`, `$_FILES['files']['tmp_name']`, etc.

d) Using a special function to parse the `$_FILES` array.

Answer: c) Through a loop iterating over `$_FILES['files']['name']`, `$_FILES['files']['tmp_name']`, etc.

- A significant security risk associated with using the original uploaded filename for storage is:

a) Potential filename collisions.

b) Exposure of the user's local file structure.

c) The possibility of executing malicious code if the filename is crafted maliciously.

d) All of the above.

Answer: d) All of the above

- What is the primary goal of Object-Oriented Programming (OOP)?

a) To write code that is only functional.

b) To organize code into reusable and manageable units called objects.

c) To execute code in a sequential manner.

d) To avoid using functions.

Answer: b) To organize code into reusable and manageable units called objects.

- Which of the following is a core principle of OOP?

a) Procedural programming

b) Functional decomposition

c) Encapsulation

d) Imperative programming

Answer: c) Encapsulation

- In OOP, what is a class?

a) An instance of an object.

b) A blueprint or template for creating objects.

c) A specific value assigned to an object's property.

d) A function that belongs to an object.

Answer: b) A blueprint or template for creating objects.

- In OOP, what is an object?

a) A blueprint for creating classes.

b) A specific instance of a class.

c) A characteristic or attribute of a class.

d) An action that a class can perform.

Answer: b) A specific instance of a class.

- Which keyword is used in PHP to define a class?

a) `class`

b) `object`

c) `new`

d) `function`

Answer: a) `class`

- Which keyword is used in PHP to create an instance of a class (an object)?

a) `class`

b) `object`

c) `new`

d) `instanceof`

Answer: c) `new`

- In PHP OOP, what are properties?

a) Actions that an object can perform.

b) Characteristics or attributes of an object.

c) Blueprints for creating objects.

d) Ways to control access to object members.

Answer: b) Characteristics or attributes of an object.

- In PHP OOP, what are methods?

a) Characteristics or attributes of an object.

b) Blueprints for creating objects.

c) Functions that belong to a class and can operate on its objects.

d) Keywords used to define classes.

Answer: c) Functions that belong to a class and can operate on its objects.

- Which of the following is NOT a common visibility modifier for class properties and methods in PHP?

a) `public`

b) `private`

c) `protected`

d) `internal`

Answer: d) `internal`

- What does the `public` visibility modifier allow?

a) Access only from within the class itself.

b) Access from within the class and its parent classes.

c) Access from anywhere (within the class, its parent classes, and outside the class).

d) No access from outside the class.

Answer: c) Access from anywhere (within the class, its parent classes, and outside the class).

- What does the `private` visibility modifier allow?

a) Access from anywhere.

b) Access only from within the class itself.

c) Access from within the class and its parent classes.

d) No access at all.

Answer: b) Access only from within the class itself.

- What does the `protected` visibility modifier allow?

a) Access only from outside the class.

b) Access only from within the class.

c) Access from within the class and its parent classes.

d) No access from within the class.

Answer: c) Access from within the class and its parent classes.

- What is inheritance in OOP?

a) The ability of an object to contain other objects.

b) The mechanism by which a class can acquire the properties and methods of another class.

c) The process of hiding the internal implementation details of an object.

d) The ability of objects of different classes to respond to the same method call in their own way.

Answer: b) The mechanism by which a class can acquire the properties and methods of another class.

- Which keyword is used in PHP to implement inheritance?

a) `implements`

b) `extends`

c) `uses`

d) `inherits`

Answer: b) `extends`

- What is polymorphism in OOP?

a) The ability of a class to have multiple constructors.

b) The mechanism of hiding the internal details of an object.

c) The ability of objects of different classes to respond to the same method call in their own way.

d) The process of creating new classes from existing ones.

Answer: c) The ability of objects of different classes to respond to the same method call in their own way.

- What is encapsulation in OOP?

a) The ability of a class to inherit from multiple parent classes.

b) The mechanism of bundling data (properties) and the methods that operate on that data within a single unit (class).

c) The ability of objects to take on many forms.

d) The process of creating objects.

Answer: b) The mechanism of bundling data (properties) and the methods that operate on that data within a single unit (class).

- What is an abstract class in PHP?

a) A class that cannot have any methods.

b) A class that can be directly instantiated.

c) A class that can have abstract methods (declared but not implemented) and cannot be directly instantiated.

d) A class that automatically inherits from all other classes.

Answer: c) A class that can have abstract methods (declared but not implemented) and cannot be directly instantiated.

- Which keyword is used to define an abstract method in PHP?

a) `static`

b) `final`

c) `abstract`

d) `interface`

`Answer:` c) `abstract`

- What is an interface in PHP?

a) A class that can only contain constants.

b) A blueprint for classes, defining a set of methods that a class implementing the interface must define.

c) An abstract class with no methods.

d) A special type of object.

Answer: b) A blueprint for classes, defining a set of methods that a class implementing the interface must define.

- Which keyword is used in PHP for a class to implement an interface?

a) `extends`

b) `implements`

c) `uses`

d) `inherits`

Answer: b) `implements`