MySQL Code-Based Questions (10 Questions)

1. What is the output of this MySQL query?

```
SELECT 10 / 3;
```

a) 3
b) 3.3333
c) 3.0000
d) Error

---

2. What will this query return?

```
SELECT CONCAT('5' + 2, '3' + 1);
```

a) 73
b) 83
c) 53
d) Error

---

3. What will this query do?

```
UPDATE users SET name = NULL WHERE id = 1;
SELECT name FROM users WHERE id = 1;
```

a) Returns `NULL`
b) Returns empty string
c) Throws an error
d) Returns previous value

---

4. What will happen when you execute this query?

```
SELECT COUNT(*) FROM employees WHERE salary > ALL (SELECT salary FROM
employees WHERE department = 'HR');
```

a) Counts employees with a salary higher than HR's highest salary
b) Counts employees with a salary equal to HR's highest salary

c) Counts employees with a salary lower than HR's highest salary
d) Returns `NULL`

---

5. What does this query return?

```
SELECT IFNULL(NULL, 'Default Value');
```

a) NULL
b) 'Default Value'
c) 0
d) Error

---

6. What happens in this query?

```
SELECT 1 = '1', '1' = '01', '1' = 1;
```

a) `1, 0, 1`
b) `1, 1, 1`
c) `0, 1, 0`
d) Error

---

7. What is the result of this query?

```
SELECT NOW() = SYSDATE();
```

a) `TRUE` always
b) `FALSE` always
c) Can be `TRUE` or `FALSE` depending on execution time
d) Error

---

8. What does this query do?

```
SET @x = 0;
SELECT @x := @x + 1 FROM users;
```

a) Increments `@x` for each row
b) Sets `@x` to `1` for each row

c) Returns an error
d) Sets `@x` to the total row count

---

9. What will this query return?

```
SELECT ROUND(2.5), ROUND(-2.5);
```

a) `3, -3`
b) `2, -2`
c) `3, -2`
d) `2, -3`

---

10. What does this query do?

```
SELECT * FROM users WHERE id IN (NULL);
```

a) Returns all rows
b) Returns no rows
c) Returns rows where `id` is `NULL`
d) Throws an error

Git Code-Based Questions (10 Questions)

11. What does this command do?

```
git reset --soft HEAD~1
```

a) Deletes the last commit permanently
b) Moves HEAD back one commit but keeps changes staged
c) Moves HEAD back and discards changes
d) Does nothing

---

12. What does this command return?

```
git diff HEAD~1 HEAD
```

a) Changes between the last two commits
b) Changes between working directory and last commit

c) Changes in staged files
d) Returns nothing

---

13. What will this Git command do?

```
git stash pop
```

a) Restores the last stashed change and deletes it
b) Restores all stashed changes
c) Deletes the last stash without applying it
d) Throws an error

---

14. What does this Git command output?

```
git log --oneline -n 3
```

a) Last 3 commits with full details
b) Last 3 commits in one line each
c) Commits from 3 different branches
d) Only the latest commit

---

15. What does this Git command do?

```
git branch feature && git checkout feature
```

a) Creates and checks out `feature` branch
b) Only creates `feature` branch
c) Only checks out `feature` branch
d) Throws an error

---

16. What does this Git command do?

```
git rebase master
```

a) Merges `master` into the current branch
b) Moves the current branch commits on top of `master`
c) Deletes `master` branch
d) Returns an error

---

17. What does this Git command do?

```
git fetch origin
```

a) Downloads latest changes but does not merge
b) Merges remote changes automatically
c) Deletes local commits
d) Stages changes

---

18. What does this Git command do?

```
git checkout --
```

a) Discards local changes in working directory
b) Switches branches
c) Creates a new branch
d) Merges branches

---

19. What does this Git command do?

```
git cherry-pick abc123
```

a) Applies commit `abc123` to current branch
b) Deletes commit `abc123`
c) Moves HEAD to `abc123`
d) Throws an error

---

20. What does this Git command do?

```
git push origin HEAD:refs/for/master
```

a) Creates a new branch `for/master`
b) Pushes current branch to `master`
c) Pushes a patch for code review
d) Deletes `master` branch

Core Java Code-Based Questions (10 Questions)

21. What does this Java code output?

```
System.out.println(10 + "20");
```

a) `30`
b) `1020`
c) Compilation error
d) Runtime error

---

22. What does this Java snippet do?

```
String s1 = "hello";
String s2 = new String("hello");
System.out.println(s1 == s2);
```

a) `true`
b) `false`
c) Compilation error
d) Runtime error

---

23. What does this Java snippet do?

```
int arr[] = new int[5];
System.out.println(arr[0]);
```

a) `0`
b) Compilation error
c) Runtime error
d) `null`

---

24. What will this Java code output?

```
System.out.println(5 / 0);
```

a) `Infinity`
b) `NaN`
c) Compilation error
d) ArithmeticException

---

25. What does this Java code output?

```
Integer x = 1000;
Integer y = 1000;
System.out.println(x == y);
```

a) `true`
b) `false`
c) Compilation error
d) Runtime error

Here are the **answer keys** for the **30 tricky code-based MCQs** on **MySQL, Git, and Core Java**:

MySQL Code-Based Questions Answers

1. **(b)** `3.3333` (MySQL performs floating-point division).
2. **(b)** `83` (`'5' + 2` is treated as `7`, `'3' + 1` is treated as `4`, result is `73`).
3. **(a)** Returns `NULL` (`NULL` explicitly assigned).
4. **(a)** Counts employees with a salary higher than HR's highest salary.
5. **(b)** `'Default Value'` (`IFNULL` replaces `NULL` with the second argument).
6. **(b)** `1, 1, 1` (`MySQL` treats `'1'` and `'01'` as numbers).
7. **(c)** Can be `TRUE` or `FALSE` depending on execution time (`SYSDATE()` is real-time).
8. **(a)** Increments `@x` for each row (`@x := @x + 1` increases for each row in `users`).
9. **(c)** `3, -2` (`ROUND(2.5)` rounds up, `ROUND(-2.5)` rounds down).
10. **(b)** Returns no rows (`NULL` in `IN` always results in `FALSE`).

Git Code-Based Questions Answers

11. **(b)** Moves `HEAD` back one commit but keeps changes staged (`--soft` does not delete changes).
12. **(a)** Changes between the last two commits.
13. **(a)** Restores the last stashed change and deletes it.
14. **(b)** Last 3 commits in one line each (`--oneline` shows short commit hashes).
15. **(a)** Creates and checks out `feature` branch.
16. **(b)** Moves the current branch commits on top of `master`.
17. **(a)** Downloads latest changes but does not merge (`fetch` only updates remote-tracking branches).
18. **(a)** Discards local changes in the working directory.

19. **(a)** Applies commit `abc123` to the current branch (`cherry-pick` replays commits).
20. **(c)** Pushes a patch for code review (`refs/for/master` is used in Gerrit).

Core Java Code-Based Questions Answers

21. **(b)** `"1020"` (`10` is concatenated with `"20"` as a string).
22. **(b)** `false` (`new String("hello")` creates a different object than the string pool).
23. **(a)** `0` (Array elements are initialized to `0` by default).
24. **(d)** `ArithmeticException` (`int` division by zero is not allowed).
25. **(b)** `false` (`Integer` objects outside `-128` to `127` range are different instances).
