INTRODUCTION TO SOFTWARE SYSTEMS: QUIZ 1 – SET B ANSWER KEY

Max Marks: 80 (20 × 4)

Correct options are highlighted.

1. Which command creates a copy of an existing git repository?
   a) `git copy` b) `git replace` c) `git move` d) **`git clone`**

   Explanation:
   - **`git clone`**: Correct. Creates a new local copy of an existing repository, including its history, branches, and configuration.
   - `git replace`: Used to replace objects in the Git object database for advanced history rewriting; it does not copy repositories.
   - `git move`: Not a Git command; Git uses `git mv` to rename or move files within a repository.
   - `git copy`: Not a valid Git command.

2. Which of the following character is not used for comments in C, Python or Bash?
   a) /* b) # c) **`!`** d) //

   Explanation:
   - `#`: Used for comments in Python and Bash.
   - `/*`: Used for block comments in C.
   - `//`: Used for single-line comments in C.
   - `!`: Not used for comments in any of these languages; in Bash, it is used for history expansion or in the shebang line.

3. If `file1` exists in the current directory, running

   ```
   touch file1
   ```

   will: a) Delete file1 b) Make file1 empty c) **Modify its date** d) NOTA

   Explanation: `touch` updates the access and modification timestamps of an existing file; file contents remain unchanged.

4. In Bash, the type of a variable input using a `read` statement is?
   a) **Can vary** b) int c) string d) float

   Explanation: The `read` command reads input as text and stores it as text, but Bash variables are untyped. This text is not a typed *string*; its interpretation (numeric or textual) depends entirely on how the variable is used.

```
read x
echo $x          # treated as string
echo $((x + 1))  # treated as integer
```

5. Which of the following specifies the block of statements in a loop in Bash?
   a) **do, done** b) then, fi c) begin, end d) NOTA

   Explanation:
   - (a) begin, end: Not used in Bash. These keywords are associated with other languages (e.g., Pascal) but have no syntactic meaning in shell scripting.
   - **(b) do, done**: In Bash, loop bodies (such as `for`, `while`, and `until`) are enclosed between `do` and `done`.
   - (c) then, fi: Used for conditional statements (`if-then-fi`) in Bash, not for loops.

6. Which of the following is NOT a core data type in Python programming?
   a) List b) Complex c) Float d) **Array**

   Explanation: Python's core built-in data types include lists, complex numbers, and floats. Arrays are not a core data type; they are provided by external modules.

```
x = [1, 2, 3]   # List
y = 3.14        # Float
z = 2 + 3j      # Complex
```

7. When executing a command in shell, we can append the output to an existing file using:
   a) `>>` b) `>` c) `2>` d) NOTA

   Explanation: In the shell, the operator `>>` appends standard output (`stdout`) to an existing file instead of overwriting it.
   - `>>`: Appends standard output (`stdout`) to a file.
   - `>`: Redirects standard output (`stdout`) to a file, overwriting its contents.
   - `2>`: Redirects standard error (`stderr`) to a file.

8. If $x = 2$, what will be the value of x after the following Python code?

```
x << 2
```

a) 8 b) 4 c) **2** d) 1

Explanation: TRICK QUESTION! The expression `x << 2` is evaluated, but its result is not assigned back to `x`. The value of `x` changes only if the result is explicitly reassigned, e.g., `x = x << 2`.

9. What will be the value of the following Python expression? $5 + 4$
a) 0 b) 1 c) 2 d) **9**

Explanation: bro 😭
This is an arithmetic expression, so it evaluates to a number, not a boolean value.

10. If `A=[[1,2,3],[3,4,5],[5,6,7]]`, which of the following results in a value of 6?
a) `A[-1][2]` b) `A[3][2]` c) **`A[2][1]`** d) `A[2][3]`

Explanation: Python uses zero-based indexing. `A[2]` selects the third row `[5,6,7]`, and index `1` selects the second element, which is `6`.

- `A[-1][2]`: Selects the last row `[5,6,7]` and the last element, which is `7`, not `6`.
- `A[3][2]`: Out of bounds.
- `A[2][3]`: Out of bounds.

11. In Python, the value of

```
print("xyyzxyzxzxyy".count('zx'))
```

is: a) 1 b) 2 c) **3** d) NOTA

```
>>> print("xyyzxyzxzxyy".count('zx'))
3
```

12. What will be the result of the following Python expression? `"a"+"bc"`
a) **abc** b) a c) bca d) bc

```
>>> "a" + "bc"
'abc'
```

13. What will be the output of the following Python code? If $x =$ `'abcd'`

```
for i in range(len(x)): print(i, end=' ')
```

a) error b) p q r s c) 1 2 3 4 d) **0 1 2 3**

```
>>> x = 'abcd'
>>> for i in range(len(x)): print(i, end=' ')
...
0 1 2 3 >>>
```

Explanation: In Python, if a loop or conditional has only a single statement, it can be written on the same line after the colon without indentation. The loop runs over `range(len(x)) = range(4)`, printing the indices.

14. If $S =$ `"Hyderabad"` in Python, the value of `S[3:-3]` will be:
a) dera b) **era** c) der d) NOTA

Explanation: Python slicing follows the form `S[start:end]`, where the start index is inclusive and the end index is exclusive. Negative indices count from the end of the string. Thus, characters from index `3` up to (but not including) index `-3` are selected, resulting in `"era"`.

```
>>> S = "Hyderabad"
>>> S[3:-3]
'era'
```

15. What arithmetic operators CANNOT be used with strings in Python?
a) `*` b) `-` c) `+` d) All of the above

Explanation: Python allows some arithmetic-like operators on strings, but not all.

- `*`: Can be used to repeat strings, e.g., `"ab" * 3` gives `"ababab"`.
- `-`: Cannot be used with strings; subtraction is undefined for string objects.
- `+`: Can be used to concatenate strings, e.g., `"a" + "bc"` gives `"abc"`.

16. Which of the following is a Python list?
a) `1, 2, 3` b) **`[1, 2, 3]`** c) `(1, 2, 3)` d) `{1, 2, 3}`

Explanation: Square brackets `[ ]` denote lists in Python; commas alone form a tuple, parentheses form a tuple, and braces form a set.

17. Which of the following concepts is NOT a part of Python?
    a) Lists b) Dynamic Types c) Modules d) **Pointers**

    Explanation: Python abstracts memory management and does not provide pointer operations, unlike C or C++. In contrast, lists are built-in data structures, dynamic typing is a core language feature, and modules are fundamental to Python's program organization.

18. In a bash script, we can access the first command line argument using:
    a) **$1** b) $args[0] c) $args[1] d) $#

    Explanation: Bash provides positional parameters to access command-line arguments, where `$1` refers to the first argument.
    - `$1`: Correct. Refers to the first command-line argument passed to the script.
    - `$args[0]`: Invalid in Bash; this syntax is used in other languages, not in shell scripting.
    - `$args[1]`: Also invalid for the same reason; Bash does not have an `args` array by default.
    - `$#`: Refers to the total number of command-line arguments, not an individual argument.

19. After executing `C="IIIT H"` in Python, which of the following will correct it to `"IIITH"`?
    a) `C[3:4]='T'` b) `C[3:3]='T'` c) `C.insert('T', 3)` d) **NOTA**

    Explanation: Strings in Python are immutable, meaning their contents cannot be modified in place.
    - `C[3:4] = 'T'`: Invalid because slice assignment is not allowed on strings.
    - `C[3:3] = 'T'`: Also invalid for the same reason; strings do not support item or slice assignment.
    - `C.insert('T', 3)`: Invalid because `insert` is a list method, not a string method.

20. `git fetch + git merge` is equal to?
    a) `git branch` b) **`git pull`** c) `git push` d) NOTA

    Explanation: `git fetch` downloads updates from the remote repository and updates the remote-tracking branches without modifying the current branch. `git merge` then integrates those fetched changes into the current branch.
    - **`git pull`**: Correct. It performs a `git fetch` followed by a `git merge` in a single command.
    - `git branch`: Used to create, list, or delete branches; it does not retrieve or integrate remote changes.
    - `git push`: Sends local commits to a remote repository; it does not fetch or merge remote updates.