CSL100: Introduction to Programming – Quiz-Final

Indian Institute of Technology Bhilai

Part I- Attempt any 4

Q1.Fractional Knapsack (5 marks)

You are given n items, each with profit p_i and weight w_i, and a knapsack of capacity W. Write a Python program to compute the **maximum profit** using the **fractional knapsack** approach.**Items can be selected fractionally.**

Sample Input

- **Items** (p_i, w_i): $(60, 10), (100, 20), (120, 30)$
- **Capacity** (W): $W = 50$

Expected Output (Maximum Profit)

```
Maximum Profit: 240.00
```

Q2. Recursion: Fibonacci (5 marks)

Write a recursive function `fib(n)` that returns the n-th Fibonacci number assuming:

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2},\ n \geq 2$$

Q3. Finance: Monthly EMI Calculator (5 marks)

Write a Python program to compute the **monthly EMI** for a loan.

Inputs:

- Principal P (Rupees)
- Annual interest rate R (in %) (SIMPLE INTEREST)
- Tenure T in years

Output: Print monthly EMI rounded to 2 decimals.

Q4. OOP: Quadratic Equation Class (5 marks)

Define a class `QuadraticEquation` representing $ax^2 + bx + c = 0$.

- Private data fields: _a, _b, _c
- Constructor initializes a, b, c
- Getters: `getA()`, `getB()`, `getC()`
- `getDiscriminant()` returns $\Delta = b^2 - 4ac$
- `getRoot1()`, `getRoot2()` return

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

Return 0 if $\Delta < 0$

I/O Rule: Read a, b, c and print roots depending on Δ:

- $\Delta > 0$: two roots
- $\Delta = 0$: one root
- $\Delta < 0$: print `The equation has no roots.`

Q5. Dictionary: Character Frequency and Top-k (5 marks)

Given a line of text and an integer k, write a Python program to:

(a) Ignore punctuation: `.,;:!?`

(b) Print the top-k most frequent characters as `character:count`

Part II

Q1. `list.append` in Python is a: (1 marks)

[✓ (a) Function]   (b) Module

Q2. Given `s = "quiz 5 final"`, write output of `s[-2:-6]`. 'aniB' (1 marks)

Q3.
```
B = True
B = ""
print(B)
```
What will be printed? '') (Empty string) (1 mark)

Q4.
```
for num in range(10, 14):
    for i in range(2, num):
        if num % i == 1:
            print(num)
            break
```
What will be printed? (1 mark)

Q5.
```
L_names = ["Harsh", "Amit", "Sahil", "Viresh"]
L_names.sort()
print(L_names)
```
What will be printed? (1 mark)

Q6.
```
L = [1, 55, 973, [4,5,10], "go", "there"]
print(L[2:5])
```
What will be printed? (1 mark)

Q7.
```
print(print("Hello"))
```
What will be printed? (1 mark)

Q8.
```
list1=(1,6,8,56,39)
list1[3]=0
print(list1)
```
What will be printed? (1 mark)

Q9.
```
def pattern2(n):
    for i in range(n, 0, -1):
        print("*" * i)

pattern2(5)
```
What will be printed? (2 mark)

Part III

Q1. Calculate Word Lengths but Ignore Words Containing Digits (5 marks)

Given a sentence string, use a list comprehension to extract the lengths of each word only if **the word does not contain any digits.**

Example:
Input: `"AI2025 systems are powerful"`
Output: `[7, 3, 8]` (The word `"AI2025"` is ignored because it contains digits.)

Q2.

(10 marks)

IIT Bhilai has different student clubs, and each club organizes different types of events. Using this idea, write a Python program that demonstrates **inheritance** and **function overriding.**

1. **Create a parent class `Club`:**
   - **Attributes:** `name`, `secretary`, and a list of `members`.
   - **Functions:**
     - `add_member(name)` – adds a student to the club
     - `organize_event()` – prints a general message such as `"The club is organizing an event."`
2. **Create two child classes that inherit from `Club`:**
   - `GDG_Student_Club`
     - Add an extra attribute: `tech_stack` (list of technologies)
     - Override `organize_event()` to show a tech meetup/workshop message
   - `Swara`
     - Add an extra attribute: `activities` (e.g., singing, dance, music)
     - Override `organize_event()` to show a cultural program message
3. **Create objects** of all three classes and call `organize_event()`. This should clearly show how the child classes **override** the parent class function.