Abcedi Ilacas138180211 ailacas1@mySeneca.ca  
Click or tap here to enter text.

**Using Mozilla Firefox is strongly recommended for this Activity because it can transform JSON responses into a human readable format.** (Raw Data > "Pretty Print") (#)*is**points for API and Time Zone questions.*

1. (5) What is sent via the API from one system to another? What is sent back?

API sends: A user's response to a system.

Sent back: The system's response to the user.

2. (5) Use api.agify.io to predict the age of a person using your given name and an ISO country code

API URL request: https://api.agify.io/?name=abcedi&country\_id=CA

JSON response: {"count":0,"name":"abcedi","age":null,"country\_id":"CA"}

3. (5) Use the time zone API request at worldtimeapi.org

API URL request: http://worldtimeapi.org/api/timezone/America/New\_York

JSON response: {"abbreviation":"EST","client\_ip":"50.101.90.76","datetime":"2023-11-15T11:06:10.508971-05:00","day\_of\_week":3,"day\_of\_year":319,"dst":false,"dst\_from":null,"dst\_offset":0,"dst\_until":null,"raw\_offset":-18000,"timezone":"America/New\_York","unixtime":1700064370,"utc\_datetime":"2023-11-15T16:06:10.508971+00:00","utc\_offset":"-05:00","week\_number":46}

4. (16)Using the above JSON data from worldtimeapi.org, fill in the JSON key / value pairs relating to the descriptions in the table below.

|  |  |  |
| --- | --- | --- |
| *See Response Schema* | JSON key | JSON value |
| UTC date/time in ISO8601 format | "utc\_datetime” | ":"2023-11-15T16:06:10.508971+00:00" |
| Unix UTC timestamp | "unixtime" | 1700064370 |
| Unix UTC to location difference | "unixtime" – ("raw\_offset" / 3600) | 1700064375 |
| Location's daylight-saving time difference | "dst\_offset" | 0 |
| Location date/time in ISO8601 format | "datetime" | "2023-11-15T11:06:10.508971-05:00" |
| How do you calculate the *location's* *timestamp* from the UTC timestamp using JSON keys? | "unixtime" + "raw\_offset" | *Calculated location timestamp value is:*  1700046370 |

**5.**  (5) How did you confirm that your location timestamp when converted to data/time was the same as the Location date/time in ISO8601 format in the JSON schema? Show your test and the result.

Assuming that this timestamp is in **seconds**:  
**GMT**: Wednesday, November 15, 2023 4:06:15 PM  
**Your time zone**: Wednesday, November 15, 2023 11:06:15 AM [GMT-05:00](https://www.epochconverter.com/timezones?q=1700064375)  
**Relative**: 6 days ago  
(1700064375/86400) + DATE(1970,1,1) ≈ 2023-11-15 16:06:15

‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬2023-11-15T16:06:15.0000000+00:00

2023-11-15T16:06:15.0000000Z

SDLC – Software Development Life Cycle 54 points = 9 points × 6 items, 75+ words each

**Determine**:

First, I thoroughly read the assignment specifications to understand the key objectives, expected outputs, and constraints. I identify what exactly is being asked of me.

To become comfortable with the scope, I break down the assignment into smaller, manageable tasks, which helps in creating a focused plan.

If any part of the assignment is unclear, I engage in preliminary research or consult with instructors or peers for clarity.

**Define:**

I analyze the problem in depth to understand all the required inputs, the necessary processing steps, and the expected outputs. This might involve creating use cases or scenarios.

I make sure that I understand the data types, algorithms, or functions that might be needed for the assignment.

I write down a list of requirements and constraints to ensure that I have a firm grasp of the assignment’s needs.

**Design**:

Before jumping into coding, I design a solution. This involves understanding the technical skills required and possibly creating pseudocode or a flowchart to document the algorithm.

There is definite value in writing all the coding comments first, as it helps me to structure my thoughts and code logically.

The process of design aids in the development process by providing a clear roadmap of what needs to be coded.

**Develop:**

Implementing the design into programming source code is a careful process. I start by writing comments and then gradually fill in the code.

For testing and debugging, I write tests to check if each part of my code works as expected and adjust as necessary.

I ensure the correctness of output by comparing it with the assignment requirements and expected results.

**Deliver**:

Managing the delivery and deployment of the project includes steps like submitting the assignment on the matrix server and ensuring it runs as expected in that environment.

When things do not work as expected, I troubleshoot and make necessary changes for a successful test.

Writing the reflection text involves thinking about the challenges I faced and how I overcame them, which is an integral part of learning.

**D'oh**:

To ensure that the in-lab work is maintainable for the at-home version, I keep track of source code files and ensure consistency between versions.

For maintainability, especially when later requirements are added, I practice good coding habits like commenting and using meaningful variable names.

This step is about being prepared to revisit and revise my code as new requirements or improvements arise.

Software Version 5 × 2 points each

A. The software is Valorant, and its current version is 7.10.

B. The first number, '7', likely represents a major version. This implies significant changes or updates since the last major version, possibly including major gameplay changes, new features, or substantial revisions that could affect the overall game experience.

The second number, '10', could indicate minor updates or revisions following the major version 7. These might include smaller feature enhancements, bug fixes, balance adjustments, or minor additions that improve the game but do not fundamentally change it.

C. Forward compatibility generally means that the software can work with future updates or systems. In the case of Valorant, this could mean that the game is designed to function with future updates without necessitating major overhauls for players to continue enjoying the game.

D. Backward compatibility refers to the software's ability to work with older versions or systems. For Valorant, this might involve ensuring that the game remains playable on systems that meet the minimum requirements, even as new updates or versions are released.

E. URL: Valorant Patch Notes 7.10 - https://playvalorant.com/en-us/news/game-updates/valorant-patch-notes-7-10/

Release Date: November 14, 2023

Description of a Change: One of the updates in patch 7.10 includes changes to the character Deadlock. The GravNet (C) ability now requires anyone caught by it to manually remove the net to remove the effect. Previously, players could remove it by moving out of the net radius or waiting for the effect to expire. This change is intended to create a more interesting interaction for both the player using Deadlock and their opponents.