**Python Tasks – Day5**

1. **What is the difference between shallow copy and deep copy?**

**Ans:**

In Shallow copy, a copy of the original object is stored and only the reference address is finally copied.

Whereas In Deep copy, the copy of the original object and the repetitive copies both are stored.

1. **You want to create a library management application, you need to store data about books, students in your program, how will you store data inside python using list? Create some lists with fake data to store information about library management.**

**Ans:**

**# List of books**

**books = [**

**{**

**"title": "Python Crash Course",**

**"author": "Eric Matthes",**

**"isbn": "9781593279288",**

**"available": True**

**},**

**{**

**"title": "The Alchemist",**

**"author": "Paulo Coelho",**

**"isbn": "9780062315007",**

**"available": True**

**},**

**{**

**"title": "To Kill a Mockingbird",**

**"author": "Harper Lee",**

**"isbn": "9780060935467",**

**"available": False**

**},**

**]**

**students = [**

**{**

**"name": "John Smith",**

**"student\_id": "123456",**

**"books\_borrowed": ["9781593279288"]**

**},**

**{**

**"name": "Alice Johnson",**

**"student\_id": "789012",**

**"books\_borrowed": []**

**},**

**{**

**"name": "Michael Brown",**

**"student\_id": "345678",**

**"books\_borrowed": ["9780060935467"]**

**},**

**]**

1. **This chapter introduced assignment statements, like spam = 10. What is the difference between an expression and a statement?**

**Ans:**

An expression evaluates a single value. A statement does not.

1. **Define following.**
   1. **Atomic data types / Primary data types**

**Ans:** the **atomic data types** are also known as **primitive data types**. They are the base-level data structures used in programming languages.

* 1. **Secondary data type / User Defined Data Type**

**Ans:**

The current web page context does not provide information about secondary data types in Python.

1. **What is UDF?**

**Ans:**

A user-defined function (UDF) is a function provided by the user of a program or environment, in a context where the usual assumption is that functions are built into the program or environment.