**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **19th may 2020** | | | | | **Name:** | **Shreetal Kalabandi** | |
| **Sem & Sec** | **6th sem ‘B’** | | | | | **USN:** | **4al17cs091** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **SSCD** | | | | | | |
| **Max. Marks** | | **30** | | **Score** | | | **22** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Python for machine learning** | | | | | | | |
| **Certificate Provider** | | | **Great learning** | | **Duration** | | | **1.5 hrs I spent** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement: 1.python program.** | | | | | | | | |
| **Status: Completed.** | | | | | | | | |
| **Uploaded the report in github** | | | | | **yes** | | | |
| **If yes Repository name** | | | | | **https://github.com/Shreetal** | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | |

|  |
| --- |
|  |
| #Write a simple Python program to implement Diffie–Hellman Key Exchange Example |
|  | # Variables Used |
|  | sharedPrime = int(input("\nEnter the value of p(shared prime)")) #23 |
|  | sharedBase = int(input("\nEnter the value of g(shared base)")) #5 |
|  |  |
|  | aliceSecret = int(input("\nEnter the value of a(alice secret)")) #6 |
|  | bobSecret = int(input("\nEnter the value of a(bob secret)")) #15 |
|  |  |
|  | # Begin |
|  | print( "\nPublicly Shared Variables:") |
|  | print( " Publicly Shared Prime: " , sharedPrime ) |
|  | print( " Publicly Shared Base: " , sharedBase ) |
|  |  |
|  | # Alice Sends Bob A = g^a mod p |
|  | A = (sharedBase\*\*aliceSecret) % sharedPrime |
|  | print( "\n Alice Sends Over Public Chanel: " , A ) |
|  |  |
|  | # Bob Sends Alice B = g^b mod p |
|  | B = (sharedBase \*\* bobSecret) % sharedPrime |
|  | print( " Bob Sends Over Public Chanel: ", B ) |
|  |  |
|  | print( "\n------------\n" ) |
|  | print( "Privately Calculated Shared Secret:" ) |
|  | # Alice Computes Shared Secret: s = B^a mod p |
|  | aliceSharedSecret = (B \*\* aliceSecret) % sharedPrime |
|  | print( " Alice Shared Secret: ", aliceSharedSecret ) |
|  |  |
|  | # Bob Computes Shared Secret: s = A^b mod p |
|  | bobSharedSecret = (A\*\*bobSecret) % sharedPrime |
|  | print( " Bob Shared Secret: ", bobSharedSecret ) |
|  |  |

