DAILY ONLINE ACTIVITIES SUMMARY

Date:	**08-07-2020**	**Name:**	**Manikya K**
Sem & Sec	**8th,A**	**USN:**	**4AL16CS050**

Online Test Summary

Subject	**Not Conducted**		
Max. Marks		**Score**	

Certification Course Summary

Course	**1) Robotic Process Automation (RPA)** **2) Introduction to ethical hacking** **3) Introduction to cyber security** **4) Introduction to Hadoop**		
Certificate Provider	**1) GUVI** **2) Great learner academy**	**Duration**	**RPA – 4 Hrs** **Ethical hacking - 6 Hrs** **Cyber Security - 7 Hrs** **Hadoop – 4 Hrs**

Coding Challenges

Problem Statement: Python code for the Grade Calculator program in action

Status: Solved

Uploaded the report in Github	**Yes**
If yes Repository name	**manikya-20**
Uploaded the report in slack	**Yes**

Online Test Details: (Attach the snapshot and briefly write the report for the same)

Certification Course Details: (Attach the snapshot and briefly write the report for the same)

Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)

1) Certification Course Details:

A) Robotic process Automation:

B) Introdution to ethical hacking:

C) Introduction to Cyber Security:

D) Introduction to Hadoop:

2) Coding Challenges:

```
1. Jack's dictionary
jack = { "name":"Jack Frost",
         "assignment" : [80, 50, 40, 20],
         "test" : [75, 75],
         "lab" : [78.20, 77.20]
       }

# 2. James's dictionary
james = { "name":"James Potter",
          "assignment" : [82, 56, 44, 30],
          "test" : [80, 80],
          "lab" : [67.90, 78.72]
        }

# 3. Dylan's dictionary
dylan = { "name" : "Dylan Rhodes",
          "assignment" : [77, 82, 23, 39],
          "test" : [78, 77],
          "lab" : [80, 80]
        }

# 4. Jessica's dictionary
jess = { "name" : "Jessica Stone",
         "assignment" : [67, 55, 77, 21],
         "test" : [40, 50],
         "lab" : [69, 44.56]
       }

# 5. Tom's dictionary
tom = { "name" : "Tom Hanks",
        "assignment" : [29, 89, 60, 56],
        "test" : [65, 56],
        "lab" : [50, 40.6]
      }

# Function calculates average
def get_average(marks):
    total_sum = sum(marks)
    total_sum = float(total_sum)
```

```
    return total_sum / len(marks)

# Function calculates total average
def calculate_total_average(students):
    assignment = get_average(students["assignment"])
    test = get_average(students["test"])
    lab = get_average(students["lab"])

    # Return the result based
    # on weightage supplied
    # 10 % from assignments
    # 70 % from test
    # 20 % from lab-works
    return (0.1 * assignment +
            0.7 * test + 0.2 * lab)


# Calculate letter grade of each student
def assign_letter_grade(score):
    if score >= 90: return "A"
    elif score >= 80: return "B"
    elif score >= 70: return "C"
    elif score >= 60: return "D"
    else : return "E"

# Function to calculate the total
# average marks of the whole class
def class_average_is(student_list):
    result_list = []

    for student in student_list:
        stud_avg = calculate_total_average(student)
        result_list.append(stud_avg)
        return get_average(result_list)

# Student list consisting the
# dictionary of all students
students = [jack, james, dylan, jess, tom]

# Iterate through the students list
# and calculate their respective
# average marks and letter grade
for i in students :
```

```
    print(i["name"])
    print("=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=")
    print("Average marks of %s is : %s " %(i["name"],
                    calculate_total_average(i)))

    print("Letter Grade of %s is : %s" %(i["name"],
    assign_letter_grade(calculate_total_average(i))))

    print()


# Calculate the average of whole class
class_av = class_average_is(students)

print( "Class Average is %s" %(class_av))
print("Letter Grade of the class is %s "
       %(assign_letter_grade(class_av)))
```