
National University of Computer & Emerging Sciences, Karachi

Final Examination - Fall 2020

CS-Department

Course Code: CL118

Course Name: Programming Fundamentals LAB

Instructor Name: Ms. Anam Hamid, Mr. Kariz Kamal,
Ms. Romasha Khurshid, Mr. Irfan Ayub, Ms. Sumaiyah Zahid

Date: January 02, 2021

Total Marks: 40

Duration: 80 minutes

Instructions:

1. Return the question paper immediately before answering it. In case of any ambiguity, you may make an assumption, but your assumption should not contradict any statement in the question paper.
2. Read each question completely before answering it.
3. Exam papers consist of two pages.
4. Create a folder for your student ID, e.g., Folder Name should be K20-1292, and save all your .C files there.
5. **Submission Process:** Type IP \\172.16.5.41 -> Programming Fundamentals 10:15 to 11:45 -> Teacher nameX -> SectionX
6. You are not allowed to use a cell phone/smartwatch and USB in the exam lab.
7. Cheating in any case will lead to F-GRADE directly as per university rules.

MAIN SCENARIO

You are working as a freelance software developer. Your client has asked you to design and build a software that will manage details of employees' work on a software. Following are the details for the software tasks:

- Employees must log their time w.r.t each task they are working on. Tasks are divided into 3 levels:
 - **Level 01:** This is the most time-taking activity (more than 8 hours).

- **Level 02:** This activity can be done in 1 day (7–8 hours).
 - **Level 03:** Multiple Level 03 activities can be done in 1 day (2–3 hours).
-

Requirements

Requirement 01

Create the following function:

```
void AssignTask(struct Emp* empPtr, int tskID)
```

This function assigns the task, with the given Task ID, to the employee. (Max 3 Tasks per Employee).

Requirement 02

Create the following function:

```
void PrintTaskTimeLog(struct Task tsk)
```

This function prints the time log for the task w.r.t. to the level of the task. Also the time logged for this task (if any).

Requirement 03

Create the following function:

```
void LogTaskTime(struct Emp* empPtr, struct Task* tskPtr, int time)
```

This function logs the time for the task. Also adds the time to the employee's logged time.

Requirement 04

Create the following function:

```
void PrintEmpTaskList(struct Emp emp)
```

This function prints the list of tasks assigned to an employee.

Requirement 05

Create the following function:

```
void SetTaskEstimation(int tskID, int estTime)
```

This function sets the time estimation for the task.

Requirement 06

Create the following function:

```
void AssignBulkTasks(int es, struct Emp empList[es], int ts, struct Task tskList[ts])
```

This function assigns the given list of tasks to the list of employees. (Keep in mind: Max 3 Tasks per Employee). Left tasks are also mentioned (if any).

Requirement 07

Create the following function:

```
void PrintEmployeeTimeDetails(struct Emp emp)
```

This function prints the time details (time assigned, time logged, remaining/excess time) for the given employee.

Requirement 08

Create the following function:

```
void RecursivelyPrintEmployeesInTxtFile(int es, struct Emp empList[es], FILE *fptr)
```

This function prints the given list of employees in the given file (using recursion).

Requirement 09

Create the following function:

```
struct Emp NewEmp()
```

This function returns a new employee with all the default values.

Requirement 10

Create the following function:

```
struct Task NewTask()
```

This function returns a new task with all the default values.

Good Luck!