

Figure 1

Foundation Certificate for Higher Education

Module: DOC 333 Introduction to Programming Principles

Module Leader: Mr. Sudarshan Welihinda

Assignment Type: Individual

Submission Date: 2023.12.11

Student ID: 20231264

Student Name: Ranuga Disansa Belpa Gamage

Student Email: [ranuga.20231264@iit.ac.lk](mailto:ranuga.20231264@iit.ac.lk)

Acknowledgement

I want to express my appreciation to those who have helped complete this report.

I have been able to complete this report because of the Academic advisor Mr. Sudarshan Welihinda for their invaluable support, mentorship, and feedback and the faculty members of Doc 333 (Introduction to Programming Principles).

Additionally, I would like to thank my family members for their unwilling encouragement and help throughout the report.

Thank you.

Table of Contents

[Introduction 4](#_Toc152912313)

[The Problem 4](#_Toc152912314)

[Algorithm 5](#_Toc152912315)

[Syntax of the algorithm steps 5](#_Toc152912316)

List of Figures

[Figure 1 0](file:///D:\University\IIT\DOC333\DOC333-CW-Sem1\Report\Pollution.docx#_Toc152908290)

[Figure 2 5](file:///D:\University\IIT\DOC333\DOC333-CW-Sem1\Report\Pollution.docx#_Toc152908291)

[Figure 3 6](file:///D:\University\IIT\DOC333\DOC333-CW-Sem1\Report\Pollution.docx#_Toc152908292)

[Figure 4 7](file:///D:\University\IIT\DOC333\DOC333-CW-Sem1\Report\Pollution.docx#_Toc152908293)

[Figure 5 8](file:///D:\University\IIT\DOC333\DOC333-CW-Sem1\Report\Pollution.docx#_Toc152908294)

# Introduction

## The Problem

The solution that is created is for the problem to maintain the details of projects the construction company named “XYZ” undertakes. The solution which is an information system is built using Python programming language. The main functionality of the information system is listed below.

# Algorithm

The solution which is implemented by Python Programming language is stated below in the form of algorithm steps and with an explanation of how each aspect of the program functions.

1. Start
2. # importing packages

* IMPORT datetime

1. # initialization of variables

* SET company\_name TO “XYZ Company”
* SET workers TO 0
* SET choice TO 0
* SET all\_projects TO []
* SET completed\_projects TO []
* SET execute TO True
* SET project\_names TO []
* SET possible\_inputs TO [“ongoing”, “completed”, “onhold”]
* SET statistics\_list TO [0] \* len(possible\_inputs)
* SET redirect\_choice TO False
* SET redirect\_to TO None

1. define function menu(redirect,to,company\_name,msg):

* SET main\_menu TO company\_name + ”””

Main Menu

1. Add a new project to existing projects.
2. Remove a completed project from existing projects
3. Add new workers to available workers group
4. Update details on ongoing projects
5. Project Statistics
6. Exit

“””

* OUTPUT “Redirecting…” if redirect is True else main\_menu
* RETURN to if redirect is True else INPUT user choice

1. define function remove\_completed\_projects(code\_of\_project,every\_project,workers\_tot,stats\_list,complete\_projects,possible\_stats):

* TRY
* SET index\_of\_project TO project\_names.index(code\_of\_project)
* SET date\_time TO datetime.datetime.now()
* SET actual\_end\_date TO date\_time.strftime(“%m/%d/%Y”)
* SET code\_of\_project,clients\_name,start\_date,expected\_end\_date,number\_of\_workers,\_,index TO every\_project[index\_of\_project]
* SET completed\_project\_details TO [code\_of\_project,clients\_name,start\_date,expected\_end\_date,number\_of\_workers,actual\_end\_date]
* SET workers\_tot TO workers\_tot + num\_of\_workers
* SET stats\_list[index] TO stats\_list[index] - 1
* SET stats\_list[possible\_stats.index(“completed”)] TO stats\_list[possible\_stats.index(“completed”)] + 1
* APPEND completed\_project\_details TO completed\_projects
* DELETE every\_project[index\_of\_project]
* DELETE project\_names[index\_of\_project]
* RETURN (True, “Successful removed completed projects.”,workers\_tot,status\_list,completed\_projects,every\_project,project\_names)
* EXCEPT Exception as e
* RETURN (False,e,workers\_tot,status\_list,completed\_projects,every\_project,project\_names)

1. define function create\_project(status\_list,index,code\_of\_project,clients\_name,start\_date,expected\_end\_date,number\_of\_workers,project\_status,workers\_tot,every\_project,stats\_list,possible\_stats)

* TRY
* SET status\_list[index] TO status\_list[index] + 1
* SET project\_date TO [code\_of\_project,clients\_name,start\_date,expected\_end\_date,number\_of\_workers,project\_status,index]
* IF project\_status EQUALS “ongoing” and number\_of\_workers <= workers\_tot
* RETURN (False,”There is not enough workers”, workers\_tot)
* IF project\_status EQUALS “ongoing”
* SET workers\_tot TO workers\_tot – number\_of\_workers
* APPEND code\_of\_project TO project\_names
* APPEND project\_data to all\_projects
* RETURN (True, “Successfully created a new project”, workers\_tot)
* EXCEPT Exception as e
* RETURN (False, e, workers\_tot)

1. define function update\_project\_details(status\_list,index,previous\_index,code\_of\_project,clients\_name,start\_date,expected\_end\_date,number\_of\_workers,project\_status,current\_workers,workers\_tot,previous\_project\_status)

* TRY
* IF number\_of\_workers > workers\_tot + (current\_workers if previous\_project\_stautus EQUALS “ongoing” else 0)
* RETURN (False, “Workers chosen are too much”, workers\_tot)
* IF project\_status EQUALS “ongoing”
* SET workers\_tot TO workers\_tot – number\_of\_workers
* IF previous\_project\_status EQUALS “ongoing”
* SET workers\_tot TO workers\_tot + current\_workers
* SET status\_list[index] TO status\_list[index] + 1
* SET status\_list[previous\_index] TO status\_list[previous\_index] – 1
* SET project\_data TO [code\_of\_project,clients\_name,start\_date,expected\_end\_date,number\_of\_workers,project\_status,index]
* SET index TO project\_names.index(code\_of\_project)
* SET all\_projects[index] TO project\_data
* RETURN (True, “Project details updated successfully”, workers\_tot)
* EXCEPT Exception as e:
* RETURN (False,e,workers\_tot)

1. define function date\_vertification(msg)

* SET date TO INPUT(msg)
* SET splitted\_date TO date.split(date[2] if len(date) > 3 else “ “)
* IF len(splitted\_date) != 3
* OUTPUT “Enter a valid format of the date…!”
* SET month TO splitted\_date[0]
* SET date TO splitted\_date[1]
* IF month > 12
* OUTPUT “Enter a valid month ! “
* RETURN date\_vertification(msg)
* IF date > 31
* OUTPUT “Enter a valid date !”
* RETURN date\_vertification(msg)
* RETURN date

1. define function project\_status\_vertification(msg,update\_status)

* SET project\_state TO INPUT(msg).replace(“ “,””).lower()
* IF project\_state NOT IN possible\_inputs:
* OUTPUT “The entered project status is incorrect”
* IF update\_status IS True
* SET statistics\_list[possible\_inputs.index(project\_state)] TO statistics\_list[possible\_inputs.index(project\_state)] + 1
* RETURN (project\_state,statistics\_list,possible\_inputs.index(project\_state))

1. define function project\_code\_vertification(msg,project\_codes)

* SET project\_code TO INPUT(msg)
* IF project\_code IN project\_codes
* OUTPUT “Project code already exists”
* RETURN project\_code\_vertification(msg,project\_codes)
* RETURN project\_code

1. define function check\_if\_int(msg)

* TRY
* RETURN int(INPUT(msg)
* EXCEPT
* OUTPUT “The msg entered was not an integer”
* RETURN check\_if\_int(msg)

1. while execute

* SET choice TO menu(redirect\_choice,redirect\_to)
* SET redirect\_choice TO False
* SET redirect\_to TO None
* IF choice EQUALS “1”
* OUTPUT company\_name + “Add a new project”
* SET code\_of\_project TO project\_code\_vertification(“Project Code : “, project\_names)
* IF code\_of\_project EQUALS “0”
* CONTINUE
* SET clients\_name TO INPUT(“Clients Name : “)
* SET start\_date TO date\_vertification(“Start Date (MM/DD/YYYY) : ”)
* SET expected\_end\_date TO date\_vertification(“Expected end date (MM/DD/YYYY)”)
* SET number\_of\_workers TO check\_if\_int(“Numbers of Workers : “)
* SET project\_status, status\_list, index TO project\_status\_vertification()
* SET save to INPUT(“Do you want to save the project (Yes/No)”)
* IF save.upper() EQUALS “YES”
* SET execution\_status, response\_msg, workers TO create\_project(status\_list,index,code\_of\_project,clients\_name,start\_date,expected\_end\_date,number\_of\_workers,project\_status,workers,all\_projects,statistics\_list,possible-Inputs)
* OUTPUT response\_msg + execution\_status
* ELSE:
* OUTPUT “The project was \*not\* saved !”
* ELSEIF choice EQUALS “2”
* OUTPUT company\_name + “Remove completed project”
* SET code\_of\_project TO INPUT(“Project Code : “)
* SET save TO INPUT(“Do you want to save the project (Yes/No) ? “)
* IF code\_of\_project IN project\_names AND save.upper() EQUALS “YES”
* SET Execution\_status,response\_msg,workers,status\_list,completed\_projects,every\_project, project\_names TO remove\_completed\_projects(code\_of\_project,all\_projects,workers,statistics\_list,completed\_projects,possible\_inputs)
* OUTPUT response\_msg + execution\_status
* ELSE
* OUTPUT “The project was not remove” IF save.upper() NOT EQUALS “YES” ELSE “The project does not exist”
* ELSEIF choice EQUALS “3”
* OUTPUT company\_name + “Add new workers”
* SET new\_no\_of\_workers TO check\_if\_int(“Number Workers to Add : “)
* IF save.upper() EQUALS “YES” AND new\_no\_of\_workers > 0
* SET workers TO workers + new\_no\_of\_workers
* ELSE
* OUTPUT “Workers must be more than 0” IF workers <= 0 ELSE “Workers were not added”
* ELSEIF choice EQUALS “4”
* OUTPUT company\_name + “Update Project Details”
* SET code\_of\_project TO INPUT(“Project Code : “)
* IF code\_of\_project.replace(“ “,””) EQUALS “0”
* CONTINUE
* SET clients\_name TO INPUT(“Clients Name : “)
* SET start\_date TO date\_vertification(“Start Date (MM/DD/YYYY) : “)
* SET expected\_end\_date TO date\_vertification(“Excepected End Date (MM/DD/YYYY) : “)
* SET number\_of\_workers TO check\_if\_int(“Numbers of Workers : “)
* SET project\_status,status\_list,index TO project\_status\_vertification()
* SET save TO INPUT(“Do you want to update the project details (Yes/No)”)
* IF save.upper() EQUALS “YES” and code\_of\_project IN project\_names
* SET current\_workers,previous\_project\_status,previous\_index TO all\_projects[project\_names.index(code\_of\_project)][4:]
* SET execution\_status,response\_msg,workers TO update\_project\_details(status\_list,index,previous\_index,code\_of\_project,client\_name,start\_date,expected\_end\_date,number\_of\_workers,project\_status,current\_workers,workers,previous\_project\_status)
* OUTPUT response\_msg + execution\_status
* ELSE
* OUTPUT “There isn’t a project with the mentioned project code”
* ELSEIF choice EQUALS “5”
* OUTPUT company\_name + “Project Statistics”
* FOR idx, item IN enumerate(possible\_inputs)
* OUTPUT “Number of “ + item + “projects : “ + statistics\_list[idx]
* OUTPUT “Number of Available Workers : “ + workers
* SET add\_project TO INPUT(“Do you want to add the project”)
* IF add\_project.upper() EQUALS “YES”
* SET redirect\_choice, redirect\_to TO True, “1”
* ELSEIF choice EQUALS “6”
* SET execute TO False
* ELSE
* OUTPUT “Please enter a valid choice!”