

# Dorrigo OS

Due date 11:59PM Monday 2 November 2020  
AEDT (local Sydney time)

## A message to all students about posting on Ed

This is an assignment and staff are not permitted to give guidance on your code, or how to solve the specific problem. That is the assessment you are required to perform to achieve the grade.

You may ask clarification questions about the assignment description. This is often necessary to implement functionality that is otherwise ambiguous.

The assignment description is not intended to be complete and you can confirm your assumptions in a form of a question. In asking the question you should be quoting the description you are asking about.

If you have a question to ask on Ed please search before asking.

*This assessment is CONFIDENTIAL. © University of Sydney.*

## Changes

1. A constraint has been relaxed on the messages of chat history. Please use the number 20 as the MAXIMUM\_CHAT\_HISTORY

file affected: dorrigo\_contact.py

nature of change: documentation

When there are more than N messages, oldest messages from this array are discarded and only the most recent N messages remain.

to

When there are more than 20 messages, oldest messages from this array are discarded and only the most recent 20 messages remain.

2. clarification of datatype file affected: dorrigo\_contact.py : update\_phone\_number

nature of change: documentation

long type

to

int type

3. clarification of maximum contacts file affected: dorrigo\_mobile.py

nature of change: documentation

| A list of 10 possible contacts

to

| A list of N possible contacts

4. clarification of `get_copy_of_owner_contact` file affected: `dorrigo_mobile.py`

nature of change: documentation

| *missing*

to

| datatype of returned object is `dorrigo_contact` or `None`

5. clarification of `set_signal_strength` file affected: `dorrigo_mobile.py`

nature of change: documentation

from

| If the phone is not connected to a network and  $n > 0$ , it will connect to a network and reduce the battery life by 2.

to

| If the phone is not connected to a network and  $n > 0$ , it will connect to a network and consequently reduce the battery life.

## Description

You are tasked with writing a component of the Dorrigo Operating System for Dorrigo Mobile. This OS will be installed in their products and distributed to many parts of the world. You are required to write the software using the Python programming language. Dorrigo OS must include support the following features within this component:

- Battery life
- Network connection
- Signal Strength
- Charging a phone
- Manage contact data
- Being able to copy contacts
- Delete contacts
- Add contacts
- Update contact details
- Copy contacts
- Manage messages
- Add messages

- Clear messages
- Get latest and oldest messages

The given properties of the class cannot be removed and their function signatures cannot be changed. You are not allowed to use:

- any imports
- for loops
- dictionaries
- several builtin functions from Python

You are provided with a scaffold and comments that describe the methods required to implement. Each method has comments to describes the processes necessary to implement.

## Factory Defaults

With every phone manufactured and installed with Dorrigio OS, the default factory settings are:

- Phone is off
- Phone has battery life (25)
- Not connected to a network
- No Signal (signal strength is 0)

## Owner contact factory default on the device

- First Name: Dorrigio
- Last Name: Incorporated
- Phone Number: 180076237867

One message should be included under this contact: "Dorrigio" says "Thank you for choosing Dorrigio products".

Stored as "Dorrigio: Thank you for choosing Dorrigio products"

The contact list should not contain ANY other contacts on first initialisation.

If the factory defaults do not match it can be suspected that the hardware contains a fault and requires inspection. By ensuring the software clearly adheres to the correctness of what is specified, this can be ruled out.

## Battery and Charging

The OS needs to keep track of the battery level and implement functions related to battery changing, charging and status. The battery level is represented as an integer between 0 and 100 inclusive. [0, 100]

- is\_phone\_on
- get\_battery\_life
- change\_battery
- charge\_phone

- use\_phone
- set\_phone\_on

Each method is described both here and in the code comments

is\_phone\_on

This method checks if the phone is on or not

get\_battery\_life

Retrieves the battery life which is represented as value between 0 and 100.

change\_battery

Changes the battery and therefore changing the battery level. The phone is switched to the off state after this operation and the battery life is updated. If the new battery's level is outside of the range accepted ( $n < 0$  OR  $n > 100$ ) then it should be rejected and no update should occur.

charge\_phone

The phone is charged and battery life increases by 10. In the event that the battery life exceeds 100, the charge becomes 100. A charge would not occur and the method should return false in the event that nothing has changed.

use\_phone

will reduce the battery level by k units of battery level. The phone will turn off if the use causes the battery level to reach 0.

set\_phone\_on

Turning the phone on will reduce the battery level by 5, if the battery level is  $< 6$  the phone should not power on.

## Network Connectivity and Signal

A baseline feature to a phone is determining the network connectivity status and updating it.

The network status has two parts. Network is connected and Signal strength. Signal strength is represented as a range between 0 and 5 inclusive [0,5]. 0 representing that the phone is not connected to a network while all numbers  $> 0$  infer that the phone is connected to a network.

- is\_connected\_network
- disconnect\_network
- connect\_network
- get\_signal\_strength
- set\_signal\_strength

Each method is described both here and in the code comments

is\_connected\_network

Reports if the phone is connected to a network

connect\_network

Connects to a network if needed otherwise does nothing. When connecting to network, sets the signal strength to 1 if the signal strength is currently set to 0. Sets the signal strength to the last known value of signal strength if it is not currently set to 0. If the network needs to connect, this process will reduce the battery life by 2.

disconnect\_network

Disconnects from a network and sets the signal strength to 0

get\_signal\_strength

Returns an integer value between 0 and 5 inclusive, that represents the signal strength

set\_signal\_strength

Sets the signal strength to  $n$ , where  $n$  must be in the range of  $[0,5]$ . If  $n$  is inside of the range of  $[0,5]$  the method will be successful.

If the phone is not connected to a network and  $n > 0$ , it will connect to a network and consequently reduce the battery life.

If the phone is connected to a network, the signal strength value is updated. If the signal strength is zero, it will disconnect the network, while a signal strength of  $> 0$  will not change the network connected status.

If  $n$  is outside of the range of  $[0,5]$ , or the phone is off, this method should not affect the mobile and specify that it did not successfully update.

## Contact Management

The OS allows for the user to manage contacts by being able to search, remove and add contacts. The maximum number of contacts that can be stored on the device is at least 10 plus the owner contact. The positive integer will be given to the operating system upon initialisation.

The following methods require to be implemented:

- search\_contact
- add\_contact
- remove\_contact
- get\_copy\_of\_owner\_contact
- get\_number\_of\_contacts

search\_contact

A user would want to find contacts that are stored on their phone. Given a name a user could use an input, the OS should check to see if the contact's first name or last name match the given input.

The method can return more than one result if the string is matched multiple times. If the phone is off, the method should not proceed to execute and instead return no entries.

add\_contact

Given a `dorrigo_contact`, the OS should add this contact to the contact list. Only when there is enough space to do so. If phone is off, the method should not add a contact and return that adding the contact failed.

`remove_contact`

Given a `dorrigo_contact`, the OS should remove this contact from the contact list. It is successful if the contact was found and removed. Otherwise failed. If phone is off, the method should not remove a contact and return that adding the contact failed. Invalid contact, such as `None`, will result in fail.

`get_number_of_contacts`

Return the number of active contacts. This is possibly less than the maximum.

`get_copy_of_owner_contact`

Returns a deep copy of the owner of this phone as a `dorrigo_contact` object. Returns `None` if the phone is in the off state.

## Contacts

A baseline feature that is required to be implemented is contact management. Each contact has a:

- First name
- Last Name
- Phone Number, cannot be less than 6 digits or greater than 14 digits.
- and Chat History

The fields First Name, Last Name and Phone Number can be updated by the user. A first and last name can be of any length and cannot be set to `None`. Each contact will have these methods associated with it:

- `get_first_name`
- `get_last_name`
- `get_phone_number`
- `update_first_name`
- `update_last_name`
- `update_phone_number`
- `create_copy`

`get_first_name`, `get_last_name`, `get_phone_number`

Retrieves the respective properties associated with the method name.

`update_first_name`, `update_last_name`, `update_phone_number`

Allows updating/changing the properties associated with the method name.

`create_copy`

This method allows a contact to be duplicated, this would be used if a contact has two phone numbers and the user would like to duplicate the user and update the phone number on one of them. This method should create a deep copy of `dorrigo_contact` object.

# Messaging

Messages are stored for each contact on the phone. Each contact can contain a maximum of 20 messages and once messages exceed that limit it will overwrite existing messages.

There are 5 methods that are required to be implemented:

- `add_chat_message`
- `get_last_message`
- `get_oldest_message`
- `clear_chat_history`

## `add_chat_message`

When a message is sent to the phone, the Dorrigo OS needs to store it and be able to retrieve it based on the contact.

The message format of a chat message when stored in the chat history is `who_said_it + ": " + message`. When two contacts communicate, the first name is `who_said_it`.

## `get_last_message`

This method should retrieve the last message from a contact, if this contact has no messages, the method should return `None`.

## `get_oldest_message`

This method should retrieve the oldest message in the chat history for a contact.

## `clear_chat_history`

Removes all messages from the chat history.

# About tests

Tests for the assignment will be released progressively. Every day up until the deadline.

There are public tests, pre-deadline hidden test cases and post-deadline hidden test cases.

- public tests show where you pass/fail and reveal some information about the nature of the test
- hidden tests show does not disclose what was tested
  - pre-deadline - shows you pass/fail
  - post-deadline - only be performed after the deadline

## An important notice

- There is no oracle in the real world
- You do not know the total number of tests
- You do not know the proportion of tests pre vs post deadline
- You will not receive tests that explore all the functionality requested from the assignment description

Test your program well.

# Report

You are to include a report about your implementation of this assignment.

The report will answer the following questions:

- What are instance variables and how do they differ from local variables? Max 200 words.
- Using `dorrigo_contact` object as an example, describe how calling an instance method differs from that of calling a function. Max 200 words.
- Describe the implementation differences between the `create_copy` method functionality if it were a shallow copy versus a deep copy operation. Max 200 words.
- Complete the table. For *all* of the instance variables used in your program. Write the class which it belongs to, the variable name, the variable datatype and whether or not it is mutable.

Sample table:

Class	Instance Variable	Datatype	Mutable
mobile	first_name	string	No
mobile	phone_number	list	Yes
mobile	...	...	...
contact	...	...	...

## Report Format

Please use the following report format for your answers to the above questions (using [markdown](#)):

SID: <SID>

unikey: <unikey>

Answer1.

Answer2.

Answer3.

Answer4.

Class	Instance Variable	Datatype	Mutable
mobile	first_name	string	No
mobile	phone_number	list	Yes
mobile	...	...	...
contact	...	...	...

A sample file has been provided for you. Please modify the SID and unikey as appropriate. A test will confirm if the report is present. It is your responsibility to ensure the report contents are correct, this is not automatically graded.



## Submission and Mark Breakdown

This assignment is 35% of your final course grade.

- Proportion of public tests passed 20%
- Proportion of hidden tests passed 10%
- Report 5%
- Total 35%

Your submission must include:

- dorrigo\_mobile.py
- dorrigo\_contact.py
- report.md
- Submit your assignment on [Ed in the Lesson](#)
- Multiple submissions are permitted.
- Only the last submission will be graded. If you continue to submit, none of the previous submissions will contribute to your grade.
- Late submissions will attract a late penalty.
- There are no marks awarded for the following test cases:
  - submitting a valid report file (testReportFileValid)
  - testMethodSignaturesCorrect
  - testConstructorFirstName