1811ICT/2807ICT/7001ICT Programming Principles, Workshop 11

School of Information and Communication Technology

Griffith University

|  |  |
| --- | --- |
| Goals | This workshop focusses on everything in the course up to classes. |
| When | Workshops from Friday 3 June to Thursday 9 June |
| Marks | 3 |
| Due | Pre-workshop questions before the start of the above mentioned workshops  Workshop programming problems by 11:59pm on 12 June |

# Before your workshop class:

* Read all of this document.
* Review the lecture notes sections 1 to 26.
* **Complete the pre-workshop questions (1 mark) posted on the course website and submit the answers for marking**.

# Workshop activities

## Problem 1

*Problem:* An Internet Service Provider (ISP) must, as part of its service to customers, provide a Domain Name Server (DNS).

A DNS is a server that maintains a database of domain names, such as www.griffith.edu.au, and their corresponding Internet Protocol Addresses (IPA), such as 132.234.243.22. A domain name has only one IPA, but an IPA may have many domain names.

Define a class that simulates a DNS. It must have:

* a method for updating the DNS with a new domain name and its IPA;
* a method for returning the IPA for a domain name, or None if it does not exist; and
* whatever private attributes are required to support the methods.

Write a test program that allows the user to test the class, by typing in fake domain names and IPAs to update the DNS, and domain names to look up. An example of the output from the test program is like this:

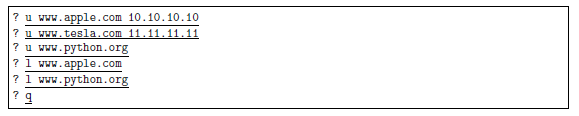
|  |
| --- |
| ? u www.google.com 8.8.8.8  ? u www.amazon.com 2.2.2.2  ? u [www.cnn.com](http://www.cnn.com)  Bad command.  ? l www.google.com  8.8.8.8  ? l www.amazon.com  2.2.2.2  ? l www.cnn.com  None  ? q |

where:

* u *DNS IPA* updates the DNS with a new domain name and its IPA;
* l *DNS* returns the IPA for a domain name, or None if it does not exist; and
* q ends the test program.

Bad inputs are to be reported and ignored.

*Answer:* Copy your code in the space given below and insert screenshots of your program output for the following scenario:



***Copy your code here***

# Matthew Prendergast

# 6th June, 2022 - Problem 1 (Workshop - Week 11)

# Define the DNS class.

class DNS:

    def \_\_init\_\_(self):

        self.dns\_dict = {}

    def update(self, domain, ip):

        self.dns\_dict[domain] = ip

    def display(self, domain):

        return self.dns\_dict.get(domain, None)

# Create the ISP object from the DNS class.

isp = DNS()

# Ptompt the user for input. Split their input into a list.

user\_int = input("? ")

command = user\_int.split()

# Process the user input.

while command[0] != "q":

    # Update the ISP disctionary.

    if len(command) == 3 and command[0] == "u":

        isp.update(command[1], command[2])

    # Retrieve the IP address for a domain.

    elif len(command) == 2 and command[0] == "l":

        print(isp.display(command[1]))

    # Bad command.

    else:

        print("Bad command.")

    # Re-prompt the user for a new input.

    user\_int = input("? ")

    command = user\_int.split()

***Insert your screenshot here***

Text

Description automatically generated

## Problem 2

*Problem:* The government now requires that DNSs should maintain a secret blacklist of IPAs that must not be returned, even if the domain name exists.

Without changing your DNS class from problem 1, define a new class that extends your old class, adding:

* a method for adding an IPA to the secret blacklist; and
* a private attribute for the blacklist. Hint: it may be called a black*list*, but is a Python list the most efficient data structure to use here?

You must also override the lookup method so that it returns None for blacklisted IPA, even if they do exist.

Write a test program that allows the user to test the new class. An example of the output from the test program is like this:

|  |
| --- |
| ? u www.google.com 8.8.8.8  ? u www.amazon.com 2.2.2.2  ? b 2.2.2.2  ? l www.google.com  8.8.8.8  ? l www.amazon.com  None  ? q |

where:

* u *DNS IPA* updates the DNS with a new domain name and its IPA;
* b *IPA* adds the IPA to the secret blacklist;
* l *DNS* returns the IPA for a domain name, or None if it does not exist or is in the secret blacklist;
* q ends the test program.

Bad inputs are to be reported and ignored.

*Answer:* Copy your code in the space given below and insert screenshots of your program output for the following scenario:



***Copy your code here***

# Matthew Prendergast

# 6th June, 2022 - Problem 2 (Workshop - Week 11)

# Define the DNS class.

class DNS:

    def \_\_init\_\_(self):

        self.dns\_dict = {}

    def update(self, domain, ip):

        self.dns\_dict[domain] = ip

    def display(self, domain):

        return self.dns\_dict.get(domain, None)

# Define the Blacklist Sub-class.

class Blacklist(DNS):

    def \_\_init\_\_(self):

        DNS.\_\_init\_\_(self)

        self.\_\_black\_set = set()

    def update\_black(self, ip):

        self.\_\_black\_set.add(ip)

    def display(self, domain):

        ip = self.dns\_dict.get(domain, None)

        if ip in self.\_\_black\_set:

            return None

        else:

            return self.dns\_dict.get(domain, None)

# Create the ISP object from the DNS class.

isp = Blacklist()

# Ptompt the user for input. Split their input into a list.

user\_int = input("? ")

command = user\_int.split()

# Process the user input.

while command[0] != "q":

    # Update the ISP disctionary.

    if len(command) == 3 and command[0] == "u":

        isp.update(command[1], command[2])

    # Retrieve the IP address for a domain.

    elif len(command) == 2 and command[0] == "l":

        print(isp.display(command[1]))

    # Update the blacklist.

    elif len(command) == 2 and command[0] == "b":

        isp.update\_black(command[1])

    # Bad command.

    else:

        print("Bad command.")

    # Re-prompt the user for a new input.

    user\_int = input("? ")

    command = user\_int.split()

***Insert your screenshot here***

Text

Description automatically generated

# Submission and marking

The pre-workshop can be accessed and submitted online using the provided link in the course website. Students get 1 mark if they get >50% in pre-workshop questions, or 0.5 mark if they get 0%-50% in pre-workshop questions, or 0 marks without any attempt.

For workshop tasks, please submit this document with copied codes and inserted screenshots using the provided submission link in the course website. Students get 2 marks if they complete both problems correctly, or 1 mark if they complete problem 1 correctly, or 0 marks without any attempt.