

# CS2102, B11

## Exam 2

---

Name:

---

You have 50 minutes to complete the problems on the following pages. There should be sufficient space provided for your answers.

If a problem asks you to create an interface, you should provide a complete interface, including method headers and argument types.

If a problem asks you to create a class:

- Include **implements** and **extends** statements
- Include field names and types
- Omit constructors
- Omit methods unless a problem asks otherwise

Omit the `Examples` class (examples of data and test cases) unless a question asks otherwise.

---

## 2. Topic: Data Structures and Methods as Arguments

A web-based email provider allows advertisers to contact users based on attributes of the user, such as their age or the words that appear frequently in their email messages. The attributes and mailboxes of each user are stored in a `User` class:

```
class User {
    String username;
    Date birthdate;
    Mailbox email;
    Words frequentWords;

    void newEmail(String message) { ... } // put message in mailbox
    int getAge() { ... } // compute age from birthdate
}
```

- (a) The `Mailbox` class contains all of a user's email messages. Which of the following data structures would you use to store the messages? Circle **one** choice and briefly justify your answer in text. For facts points on one data structure you did not choose, explain why you ruled out that choice.

List

Graph

Hashmap

Heap

Binary Tree

We need to iterate over all messages to check for info about the user

Would not use Hashmap because there isn't a clear key value.

- (b) The `Words` class maintains information on how many times various words have appeared in a user's email messages. Which of the following data structures would you use to store word-frequency information? Circle **one** choice and briefly justify your answer in text. For facts points on one data structure you did not choose, explain why you ruled out that choice.

List

Graph

Hashmap

Heap

Binary Tree

Hashmaps could map word to frequency Count

Would not use Heap because smallest frequency doesn't seem important in problem.

(exam continues next page)