

SOEN 331: Introduction to Formal Methods for
Software Engineering
Assignment 2
The Object-Z specification language

Duc Nguyen - 40064649
Vithura Muthiah - 40062305
Auvigoo Ahmed - 40128901
Ali Hanni - 40157164

*Gina Cody School of Computer Science and Software Engineering
Concordia University, Montreal, QC, Canada*

Winter 2021

Contents

1	Problem 1: State (7 pts)	3
1.1	Description:	3
1.2	Answer:	3
2	Problem 2: Class Contacts (35 pts)	4
2.1	Description:	4
2.2	Answer:	4
3	Problem 3: Class Contacts2 (8 pts)	6
3.1	Description:	6
3.2	Answer:	6

1 Problem 1: State (7 pts)

1.1 Description:

The declaration of the state of the system is defined by

- The set of phone numbers (call it *numbers*) that are recorded in contacts
 - A record of association between names and phone numbers, given by a correspondence (call it *recorded*).
1. Provide a diagram to visualize the state of the system.
 2. Provide a formal definition for numbers.
 3. Does *recorded* have to be captured by a function? What requirements would a function enforce? Explain in detail.
 4. What is the domain and the codomain of *recorded*?
 5. What type of function should *recorded* be (full or partial)? Explain in detail.
 6. Will *recorded* be an injective, surjective, or bijective? Explain in detail.
 7. Provide a formal definition for *recorded*.

1.2 Answer:

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.

2 Problem 2: Class Contacts (35 pts)

2.1 Description:

Define a formal specification in Object-Z for class *Contacts* whose interface contains the following *robust specifications*:

- **MakeNewContact**: Adds a new person to Contacts with a single phone number.
- **AddNumber**: Adds an additional phone number for an existing contact.
- **SearchForNumber**: : Returns a collection of phone numbers for a given person.
- **DeleteNumber**: Deletes an existing number.

2.2 Answer:

Contacts

$\uparrow (MakeNewContact, AddNumber, SearchForNumber, DeleteNumber)$

$numbers : \mathbb{P}PhoneNumberType$
 $recorded : PhoneNumberType \rightarrow NameType$

$numbers = ran\ recorded$

INIT

$recorded = \emptyset$

MakeNewContactOK

$\Delta()$

AddNumberOK

$\Delta()$

SearchForNumberOK

$\Xi()$

DeleteNumberOK

$\Delta()$

Success

$\Delta()$

MakeNewContact =

AddNumber =

SearchForNumber =

DeleteNumber =

3 Problem 3: Class Contacts2 (8 pts)

3.1 Description:

Subclassify **Contacts** to introduce class **Contacts2** that behaves exactly like **Contacts**, while introducing a robust operation to search for a person, given a phone number through operation **SearchForPerson**.

3.2 Answer:

