CMPS12A Introduction to Programming - Spring 2005
Programming Assignment 2

Due April 25, 2005, 9pm
(Late submissions will not be accepted.)

Objective: statements and control flow, functional abstraction.

Assignment: Write a Java program to manage your address book. For a start, your address book manager will keep only three contacts and provides the following four options:

1. Enter the name and age of a contact. If there are already three contacts in the address book, no action will be taken if this option is chosen.

2. Print out all contacts in ascending order of name.

3. Print out all contacts in ascending order of age.

4. Exit the program.

Sample runs:

> java AddressBook
1. Enter a person into addressbook
2. Print addressbook in ascending order of name
3. Print addressbook in ascending order of age
4. Exit
Enter your choice:
1
Input Contact
   Enter name of person: Andre Agassi
   Enter age of person: 35

1. Enter a person into addressbook
2. Print addressbook in ascending order of name
3. Print addressbook in ascending order of age
4. Exit
Enter your choice:
1
Input Contact
   Enter name of person: Roger Federer
   Enter age of person: 24

1. Enter a person into addressbook
2. Print addressbook in ascending order of name
3. Print addressbook in ascending order of age
4. Exit
Enter your choice:
1
Input Contact
   Enter name of person: Albert Bertossi
   Enter age of person: 28

1. Enter a person into addressbook
2. Print addressbook in ascending order of name
3. Print addressbook in ascending order of age
4. Exit
Enter your choice:
1
Address book is full.

1. Enter a person into addressbook
2. Print addressbook in ascending order of name
3. Print addressbook in ascending order of age
4. Exit
Enter your choice:
2
Contact(Name: Albert Bertossi, Age: 28)
Contact(Name: Andre Agassi, Age: 35)
Contact(Name: Roger Federer, Age: 24)

1. Enter a person into addressbook
2. Print addressbook in ascending order of name
3. Print addressbook in ascending order of age
4. Exit
Enter your choice:
3
Contact(Name: Roger Federer, Age: 24)
Contact(Name: Albert Bertossi, Age: 28)
Contact(Name: Andre Agassi, Age: 35)

1. Enter a person into addressbook
2. Print addressbook in ascending order of name
3. Print addressbook in ascending order of age
4. Exit
Enter your choice:
4

Please use the following code snippets to help you get started. The class Contact.java is complete and is given to you.

class Contact {
   private String name;
   private int age;

   public Contact(String n, int a) {
      name = n; age = a;
   }
}
public Contact() {};

public String getName() {
    return name;
}
public int getAge() {
    return age;
}

public void setName(String newname) {
    name = newname;
}
public void setAge(int newage) {
    age = newage;
}

public boolean compareName(Contact p) {
    return (name.compareTo(p.getName())<0);
}

public String toString() {
    StringBuffer sb = new StringBuffer();
    sb.append("Contact( Name: ");
    sb.append(name);
    sb.append(", ");
    sb.append("Age: ");
    sb.append(age);
    sb.append("");
    return sb.toString();
}

import tio.*;

class AddressBook {
    Contact p1, p2, p3;
    int counter = 1;

    public void inputContact() {
        String n; int a;
        ReadInput inp = new ReadInput(System.in);

        System.out.println("Input Contact");
        System.out.print(" Enter name of person: ");
        n = inp.readLine();

        System.out.print(" Enter age of person: ");
        a = inp.readInt();

        if (counter == 1) p1 = new Contact(n,a);
        else if (counter == 2) p2 = new Contact(n,a);
        else if (counter == 3) p3 = new Contact(n,a);
counter ++;
}

public void printByName() {
    if (p1.compareName(p2)) {
        // to fill in ...
    } else {
        // to fill in ...
    }
}

public void printByAge() {
    if (p1.getAge() <= p2.getAge()) {
        // to fill in ...
    } else {
        // to fill in ...
    }
}

public static void main(String[] args) {
    int choice;
    AddressBook pdb = new AddressBook();

    while (true) {
        System.out.println();
        System.out.println("1. Enter a person into addressbook");
        System.out.println("2. Print addressbook in ascending order of name");
        System.out.println("3. Print addressbook in ascending order of age");
        System.out.println("4. Exit");
        choice = Console.in.readInt();
        if (choice == 1) {
            if (pdb.counter == 4)
                System.out.println("Address book is full.");
            else pdb.inputContact();
        } else if (choice == 2) pdb.printByName();
        else if (choice == 3) pdb.printByAge();
        else if (choice == 4) break;
    }
}
Submitting your program

1. Name your source files **AddressBook.java** and **Contact.java**

2. Ensure that your program compiles and executes correctly on **unix.ic.ucsc.edu** before submitting.

3. Log onto **unix.ic.ucsc.edu** and submit the source code using the command:
   
   `submit cmps012a-wt.s05 prog2 ...
   
   You can submit more than once, for example to overwrite your previous submission with new/improved code. However, each pair should designate 1 account to be used for doing submissions.`