**<COMP\_HW1>**

2015-18634 Hyunjong Lee

1. **Programming Environment**

**OS :** Microsoft window10

**(C++)** **compiler :** MinGW GCC / **version :** C++98 / **IDE :** eclipse C/C++ IDE

**(java)** **compiler :** javac 1.8.0\_77 / **version :** jdk 1.80 / **IDE :** eclipse javaEE IDE

1. **Abstract**

This program is for managing phone book. Input value for phone book is person’s name and phone number. We could also write the team where person works, his or her birthday, and age.

I used OOP(Object-Oriented Programming) to make the program. First, I divided input values into Person, Work, Family and Friend class. Person class just contains name and phone number. Compared to Person class, Work class has the team(=office). Family class has birthday and Friend class has age. Then, I made the list of Person object to manage phone book. I could add person, remove person from phone book, and print the list. So I made PhoneBookManager class that has list of Person object and add, remove, print function for OOP.

1. **Functionality of Program**

In the main function, the program will start when I press enter key. There are three processes, “1. Add person”, “2. Remove person”, and “3. Print phone book”. Input the first key to select a process what are you going to do. If it is 1, input the second key(1~4) to choose the kind of person added to phone book. If it is 2, input the index of person who you want to remove from the phone book. If it is 3, print all person in the phone book.

1. **In depth explanation about class topics and the code**

For OOP(Object-oriented programming), I try to divide task as many as I can. Because one class

(1) Person class

Member : String first name, String last name, int phonenumber

Member function : getter and setter of three members, void print()

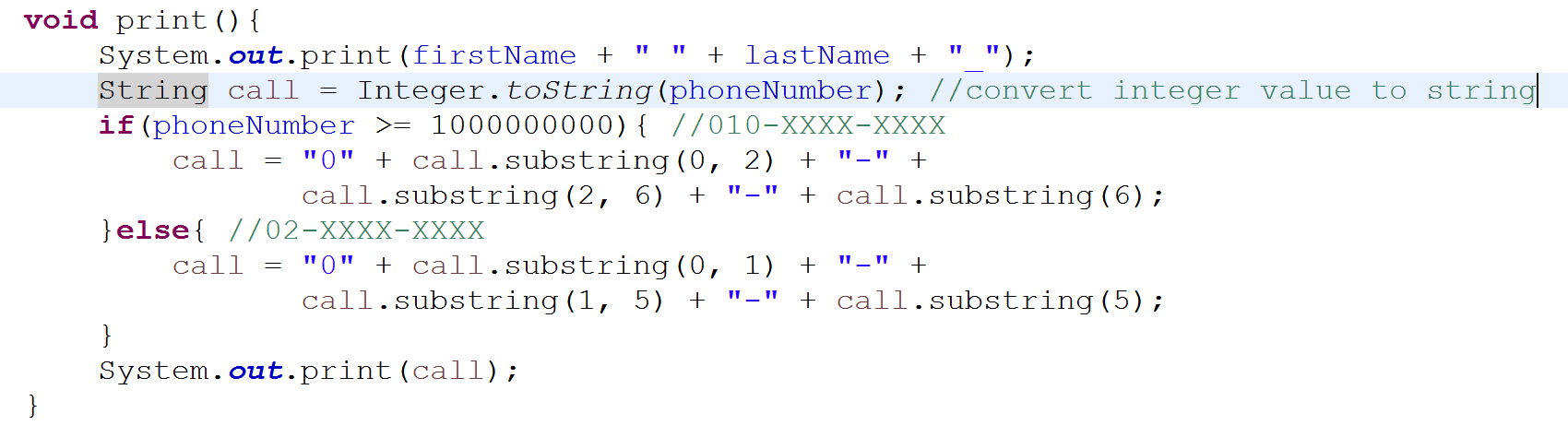


Figure Function print() of Person class implemented in java

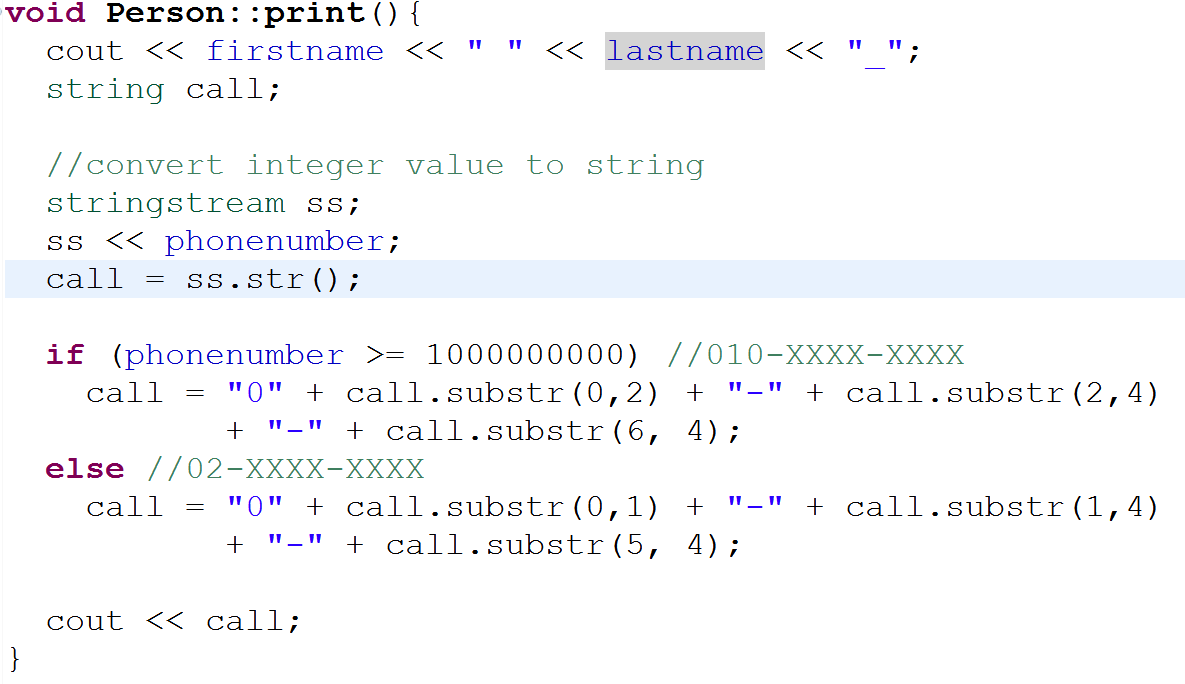


Figure Function print() of Person class implemented in C++

This code is print function in Person class. If phone number is more than 1000000000, its form is “010-XXXX-XXXX”. In other case, its form is “02-XXXX-XXXX”. When I implements this function, I need to convert integer value to string. So I used ‘Integer.toString(String s)’ function in java. In C++, I used stringstream object in header <sstream>. The integer value of phone number is stored in file stream and converted to string by str() function in stringstream.

(2) Work class

Work class is inherited Person class. Compared to parent class, String team is added to member. In java, I used ‘super’ keyword to use functions in parent class. But in C++, I could use functions in parent class by write “ : (parent class function name) (argument)” after function declaration in child class.

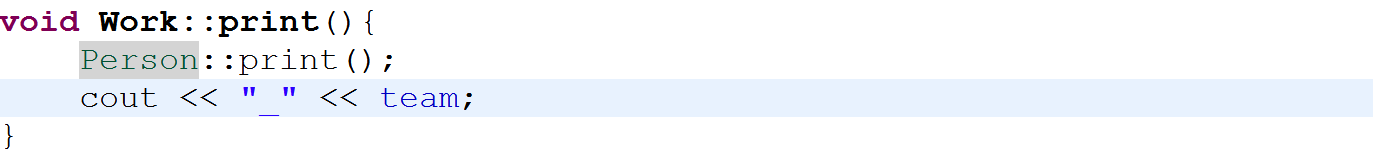
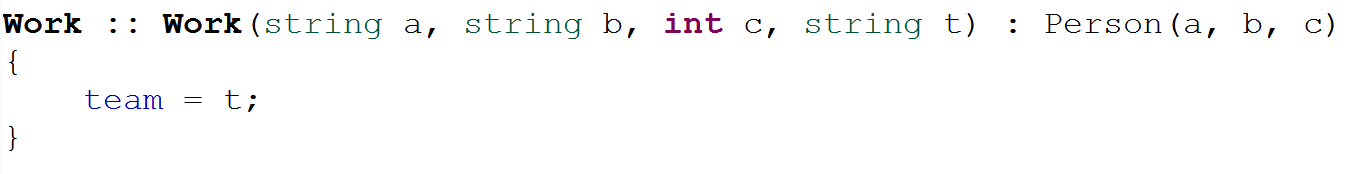


Figure The code about using the function of parent class in child class in C++

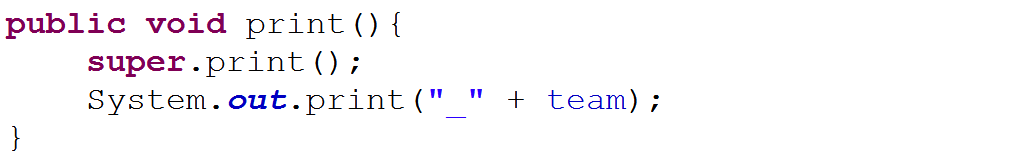
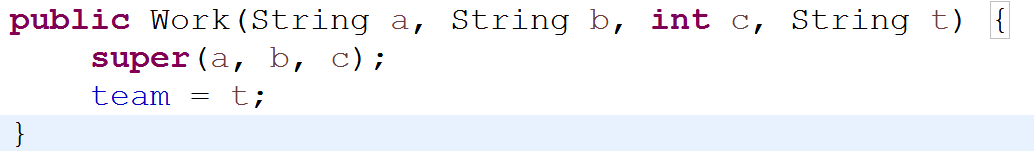


Figure The code about using the function of parent class in child class in java

(3) Friend class (inherited Person class)

It has the value of age in member.

(4) Family class (inherited Person class)

It contains the birthday and dDay() function.

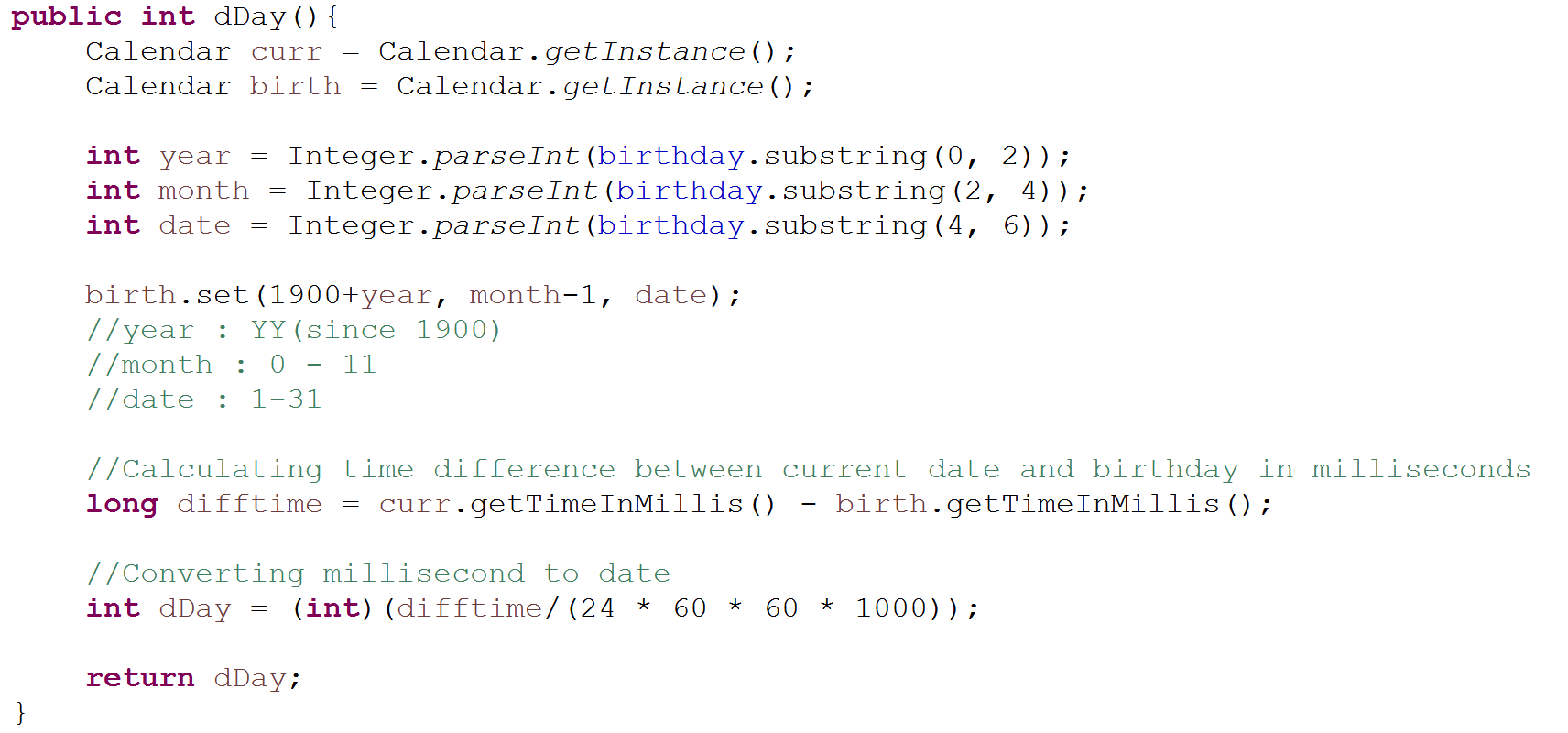


Figure Implementation of dDay() function in java

I used the Calendar class for calculating dDay. In java, we can get current time like the first line of Figure 5. Then I set Calendar object birth to birthday. We could get milliseconds between 1970 and Calendar value with getTimeInMillis() function. So I could calculate exact dDay.

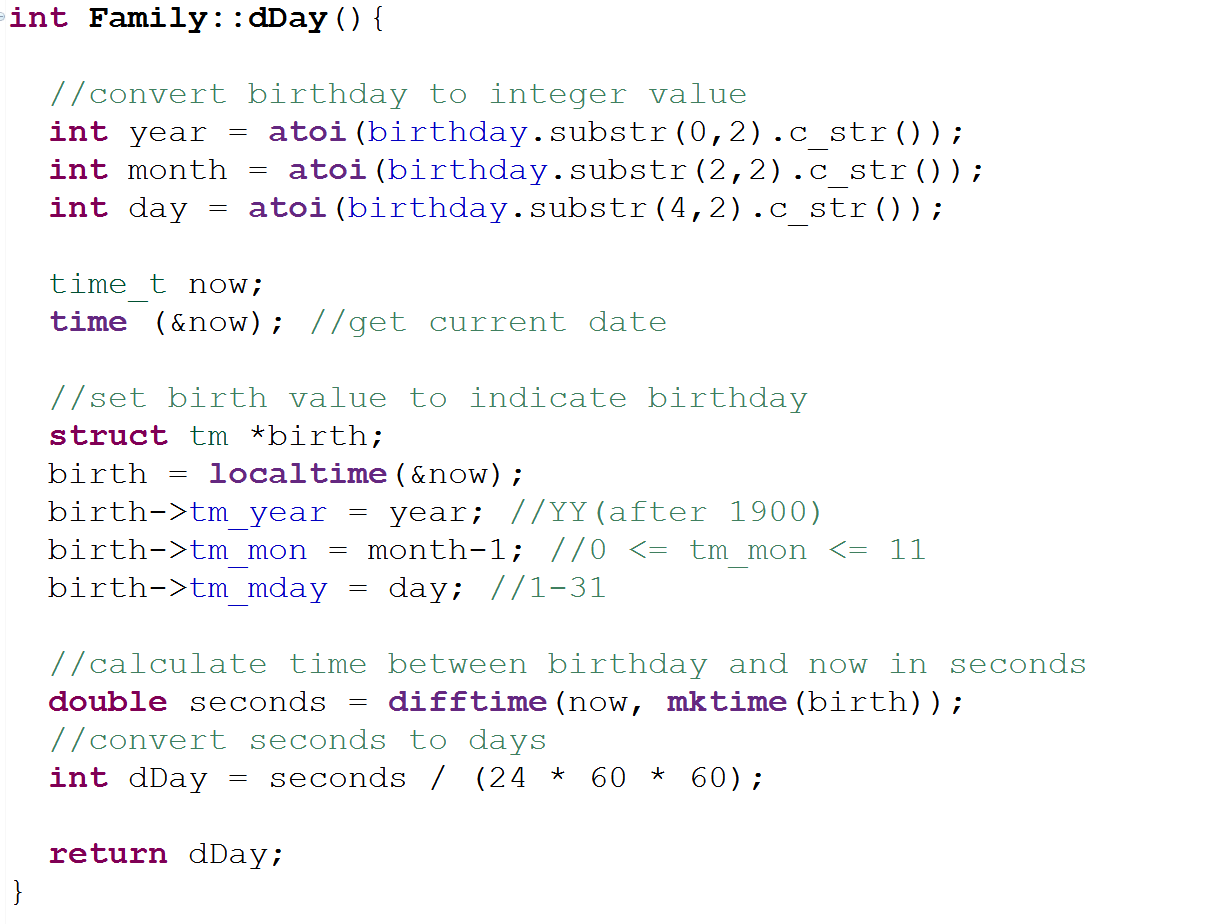


Figure Implementation of dDay() function in C++

I used ‘time\_t’ type and ‘struct tm’ in header <time.h>. we can get current time like line 7,8 in Figure 6. Algorithm is same with code in Figure 5.

(5) PhoneBookManager class

Member : List<Person> phoneBook (in java), vector<Person \*> phonebook (in C++)

Function : addCmd(int key, String firstName, String lastName, int phoneNumber), removeCmd(int index), print()

This class is for OOP programming. It could add person to phone book, remove person from phone book and print phone book.

In java, compiler consider Person object as reference value when we declare List<Person> phonebook. But in C++, we should declare phonebook as vector of Person pointer. If you declare phonebook as vector of Person object, push\_back() function is error. Because argument of push\_back function is constant reference about object. Then, we can add Person object in dynamic memory.

Also, I used iterator when I write print function. In java, I implemented code with for-each loop. In C++, I wrote code with for loop using iterator.

In C++, when I used iterator, print function in child class wasn’t called because iterator type is a person pointer not a work, family and friend pointer. So making print function virtual in person class is needed. Then when we call function of parent class, function of child class is also called.

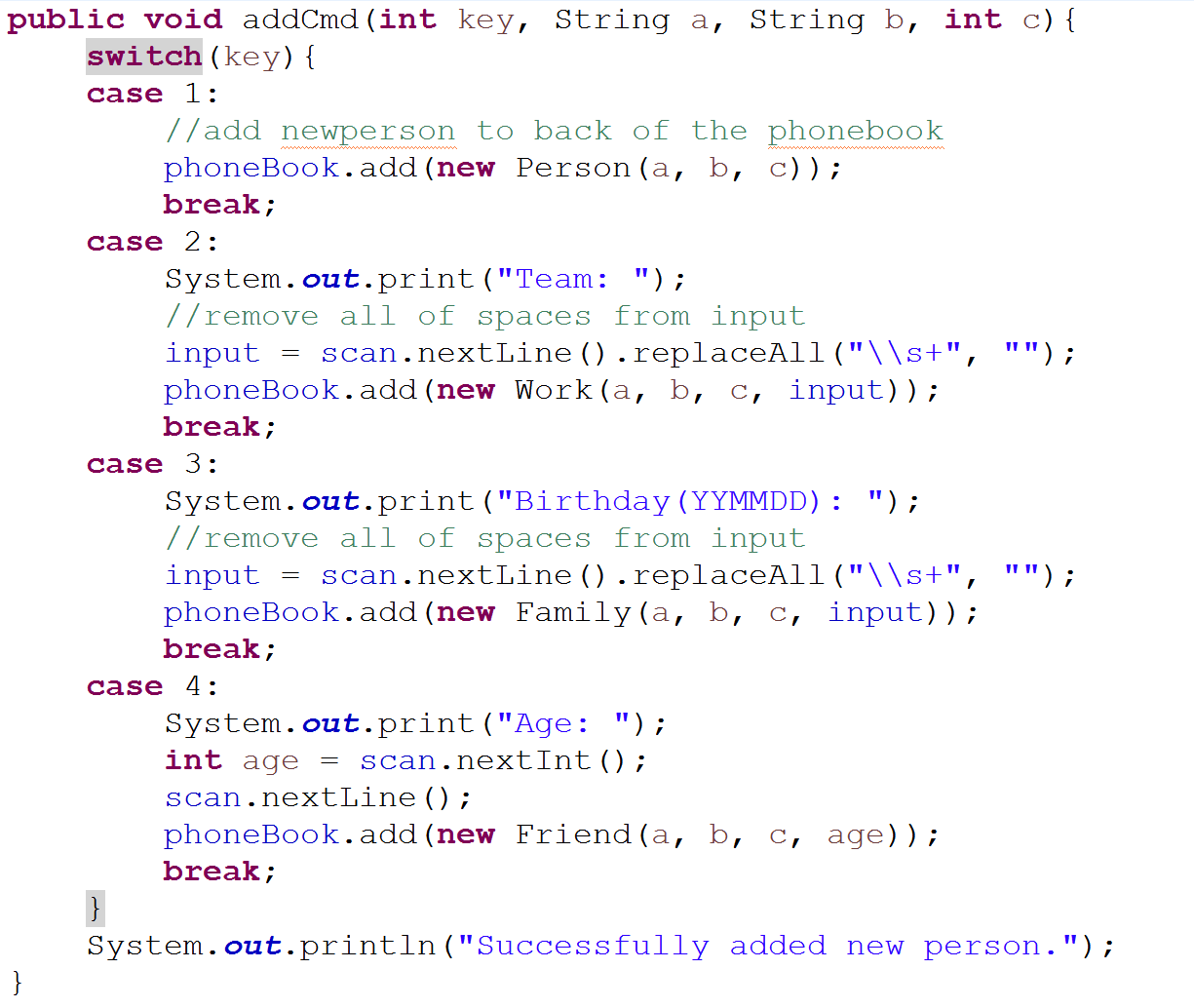


Figure addCmd function in java

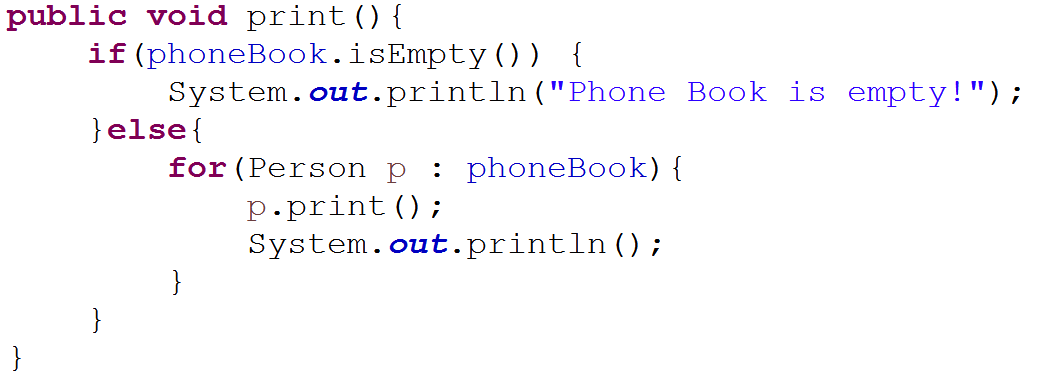


Figure print function in java

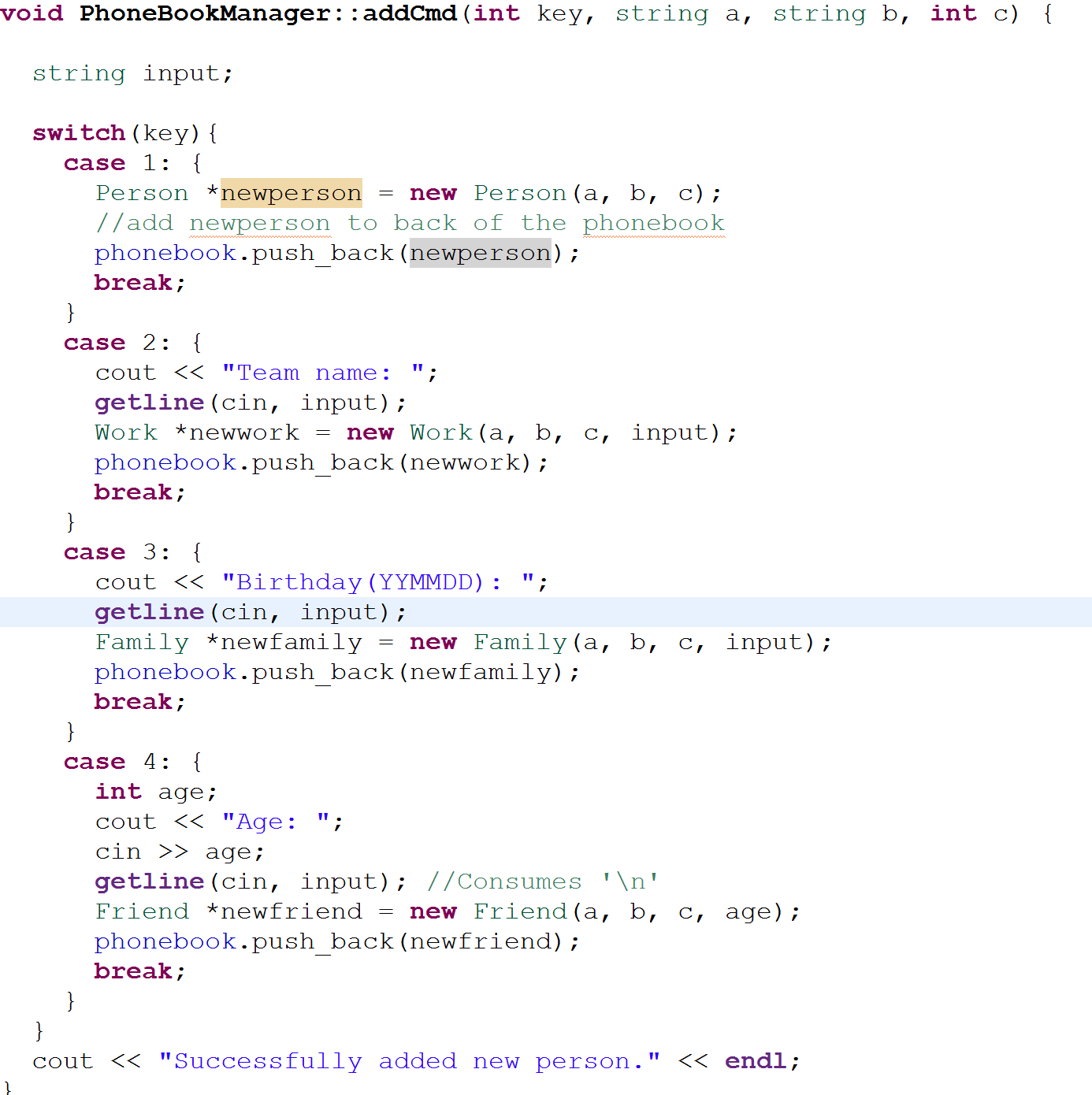


Figure addCmd function in C++

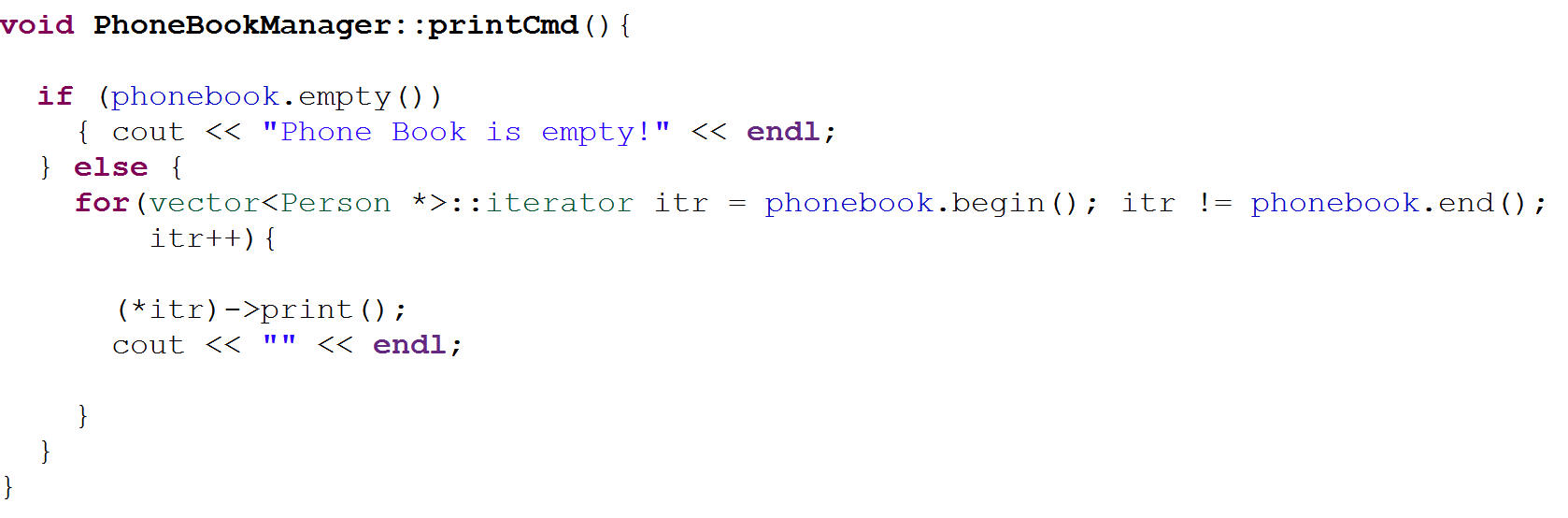


Figure print function in C++

(6) PhoneBook class

It contains main function. For splitting name into first name and last name, I used split() function in java and substr() in C++. (Split name with first blank.)

For delete character ‘-‘ from phone book, I used replaceAll() function in java and erase function in C++.

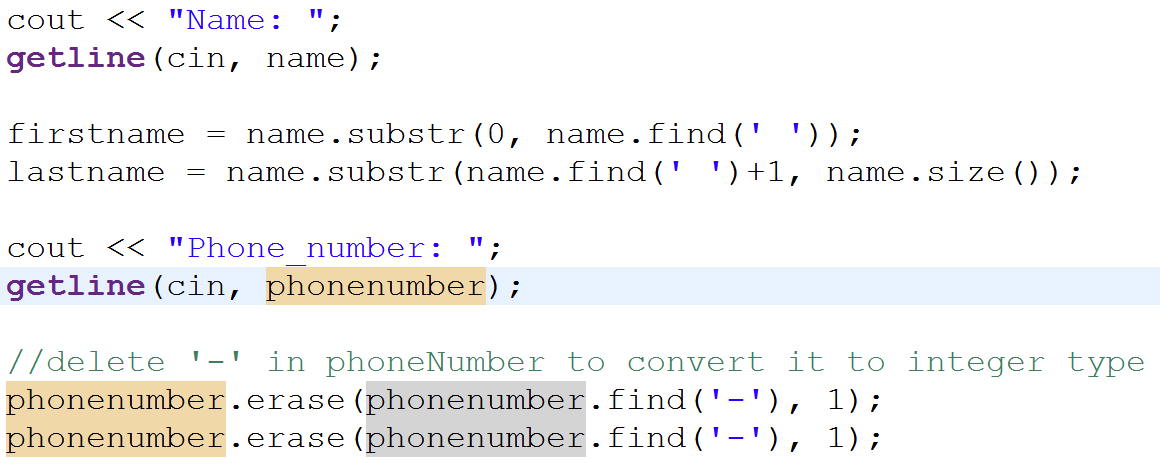


Figure manipulating input(name and phone number) in C++

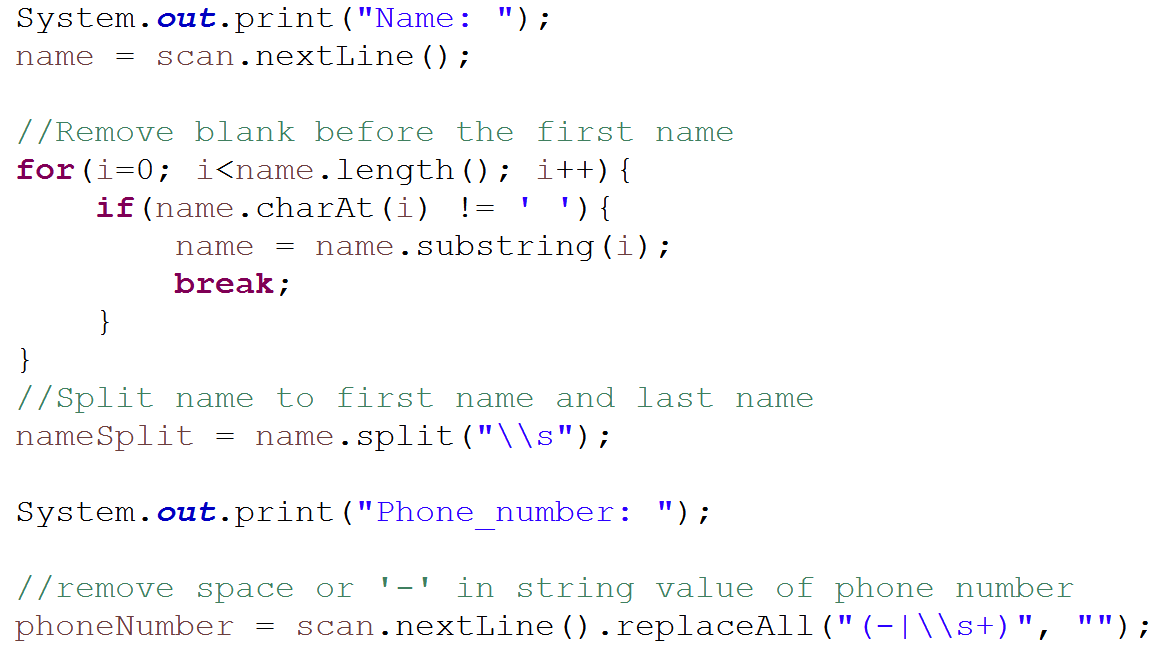


Figure manipulating input(name and phone number) in java

1. **Sample output**

(Java file)

|  |
| --- |
| 1. CP-2015-18634> 2. Phone Book 3. 1. Add person 4. 2. Remove person 5. 3. Print phone book 6. CP-2015-18634>1 7. Select Type 8. 1. Person 9. 2. Work 10. 3. Family 11. 4. Friend 12. CP-2015-18634>1 13. Name: Hyunjong Lee 14. Phone\_number: 010-0000-1111 15. Successfully added new person. 16. CP-2015-18634> 17. Phone Book 18. 1. Add person 19. 2. Remove person 20. 3. Print phone book 21. CP-2015-18634>3 22. Hyunjong Lee\_010-0000-1111 23. CP-2015-18634> 24. Phone Book 25. 1. Add person 26. 2. Remove person 27. 3. Print phone book 28. CP-2015-18634>1 29. Select Type 30. 1. Person 31. 2. Work 32. 3. Family 33. 4. Friend 34. CP-2015-18634>2 35. Name: Kyujong Lee 36. Phone\_number: 010-2222-2222 37. Team: SNU 38. Successfully added new person. 39. CP-2015-18634> 40. Phone Book 41. 1. Add person 42. 2. Remove person 43. 3. Print phone book 44. CP-2015-18634>3 45. Hyunjong Lee\_010-0000-1111 46. Kyujong Lee\_010-2222-2222\_SNU 47. CP-2015-18634> 48. Phone Book 49. 1. Add person 50. 2. Remove person 51. 3. Print phone book 52. CP-2015-18634>1 53. Select Type 54. 1. Person 55. 2. Work 56. 3. Family 57. 4. Friend 58. CP-2015-18634>3 59. Name: Sunyoung Lee 60. Phone\_number: 010-3333-3333 61. Birthday(YYMMDD): 670409 62. Successfully added new person. 63. CP-2015-18634> 64. Phone Book 65. 1. Add person 66. 2. Remove person 67. 3. Print phone book 68. CP-2015-18634>3 69. Hyunjong Lee\_010-0000-1111 70. Kyujong Lee\_010-2222-2222\_SNU 71. Sunyoung Lee\_010-3333-3333\_670409\_17908 72. CP-2015-18634> 73. Phone Book 74. 1. Add person 75. 2. Remove person 76. 3. Print phone book 77. CP-2015-18634>1 78. Select Type 79. 1. Person 80. 2. Work 81. 3. Family 82. 4. Friend 83. CP-2015-18634>4 84. Name: Beomdal Lee 85. Phone\_number: 010-4444-4444 86. Age: 52 87. Successfully added new person. 88. CP-2015-18634> 89. Phone Book 90. 1. Add person 91. 2. Remove person 92. 3. Print phone book 93. CP-2015-18634>3 94. Hyunjong Lee\_010-0000-1111 95. Kyujong Lee\_010-2222-2222\_SNU 96. Sunyoung Lee\_010-3333-3333\_670409\_17909 97. Beomdal Lee\_010-4444-4444\_52 98. CP-2015-18634>2 99. Press enter to go next step! 100. CP-2015-18634> 101. Phone Book 102. 1. Add person 103. 2. Remove person 104. 3. Print phone book 105. CP-2015-18634>2 106. Enter index of person: 2 107. A person is successfully deleted from the Phone Book! 108. CP-2015-18634> 109. Phone Book 110. 1. Add person 111. 2. Remove person 112. 3. Print phone book 113. CP-2015-18634>3 114. Hyunjong Lee\_010-0000-1111 115. Kyujong Lee\_010-2222-2222\_SNU 116. Beomdal Lee\_010-4444-4444\_52 117. CP-2015-18634>exit 118. Bye. |

(C++ file)

|  |
| --- |
| CP-2015-18634>  Phone Book  1. Add person  2. Remove person  3. Print phone book  CP-2015-18634>3  Phone Book is empty!  CP-2015-18634>  Phone Book  1. Add person  2. Remove person  3. Print phone book  CP-2015-18634>1  Select Type  1. Person  2. Work  3. Family  4. Friend  CP-2015-18634>1  Name: Kyujong Lee  Phone\_number: 010-0000-0000  Successfully added new person.  CP-2015-18634>  Phone Book  1. Add person  2. Remove person  3. Print phone book  CP-2015-18634>1  Select Type  1. Person  2. Work  3. Family  4. Friend  CP-2015-18634>2  Name: Sunyoung Lee  Phone\_number: 010-1111-1111  Team name: SNU  Successfully added new person.  CP-2015-18634>  Phone Book  1. Add person  2. Remove person  3. Print phone book  CP-2015-18634>3  Kyujong Lee\_010-0000-0000  Sunyoung Lee\_010-1111-1111\_SNU  CP-2015-18634>  Phone Book  1. Add person  2. Remove person  3. Print phone book  CP-2015-18634>1  Select Type  1. Person  2. Work  3. Family  4. Friend  CP-2015-18634>3  Name: Hyunjong Lee  Phone\_number: 02-2222-2222  Birthday(YYMMDD): 970526  Successfully added new person.  CP-2015-18634>  Phone Book  1. Add person  2. Remove person  3. Print phone book  CP-2015-18634>1  Select Type  1. Person  2. Work  3. Family  4. Friend  CP-2015-18634>4  Name: Beomdal Lee  Phone\_number: 010-3333-3333  Age: 52  Successfully added new person.  CP-2015-18634>  Phone Book  1. Add person  2. Remove person  3. Print phone book  CP-2015-18634>3  Kyujong Lee\_010-0000-0000  Sunyoung Lee\_010-1111-1111\_SNU  Hyunjong Lee\_02-2222-2222\_970526\_6905  Beomdal Lee\_010-3333-3333\_52  CP-2015-18634>  Phone Book  1. Add person  2. Remove person  3. Print phone book  CP-2015-18634>2  Enter index of person: 2  A person is successfully deleted from the Phone Book!  CP-2015-18634>  Phone Book  1. Add person  2. Remove person  3. Print phone book  CP-2015-18634>3  Kyujong Lee\_010-0000-0000  Hyunjong Lee\_02-2222-2222\_970526\_6905  Beomdal Lee\_010-3333-3333\_52  CP-2015-18634>exit  Bye. |

1. **Conclusion**

I had learned java last year, so It doesn’t matter to write java program. But I have difficult in writing program in C++ because java language supports many things than C++ like considering Person object as reference value. But I could learn a lot about C++ programming through this assignment. I learned basic of C++ programming, virtual keyword and good programming style!

I thought some part of this program should be revised while I was doing homework. In my opinion, console command is complicated. We should press enter key to do task for phone book. But it would be better to putting exit command into task for phone book(‘4. Quit Program’).