HOMEWORK 1

You will create a phone book management program using both Java and C++. For parent class, you have to implement Person class. For child classes, you have to implement Work, Friend and Family. Class Person will have three attributes; name and phone number. It will also have seven functions; set and print each attributes and print all attributes. More information about parent class Person can be found below.

```
Class Person
{
 public:
   Person( string &, string &, int &);

   setFirstName( string &); //set fistName
  string getFirstName() ; //return fistName

   setLastName(string &);
   string getLastName() ;

   setPhoneNumber(int &); //02-XXXX-XXXX or 010-XXXX-
                                                    XXXX
   int getPhoneNumber() ;
   void print();
 private:
  String fistName;
  String lastName;
  int phoneNumber;
};
```

Although every child class inherits the attributes from the parent class, each has unique attributes. You have to implement three child classes; work, friend and family, using the information provided below.

```
Class Work : public Person {

  public:
  work(string &, string &,  int &,  string &);
  void setTeam(string); //set team
  string getTeam() ; //return team
  void print() ; //print work object.   Use parent function get
attribute.

 private:

  string team; //team that person is in };



Class Friend : public Person
{
 public:

  Friend( string &,  string &, int &, int &);

  void setAge(int); //set age
  int getAge(); //return age

  void print() ; //print work object. Use parent  function
get attribute.

 private:
  int age; //age of friend
};


Class Family : public Person {

  public:

  Family(string &, string &, int &, string &);

 void setBirthday(string); //set birthday (YYMMDD)
string getBirthday() ; //return birthday

  int dDay() ; //calculate date difference between birthday and
current time

  void print() ; //print work object . Use parent function get
attribute.
private:

  string birthday; //birthday of family
};
```

After you have implemented both parent and child classes, you must define data structure to store a phone book which consists of "Person." In order to ensure dynamic allocation, you must use the vector to store phone book. In java, you must use the ArrayList which also provides dynamic allocation.

```
In C++
  vector<Person> phoneBook;

In Java
  List<Person> phoneBook = new ArrayList<Person>();
```

Since you have defined classes and the data structure, you have to fill the data structure. In order to receive information which you need to declare each object, you must have an interactive console program. The interactive console program will display appropriate prompts and receive information from users. Three functionalities that you have to implement are adding new person, Removing person and printing phone book. The following sample output will help you understand the functionalities of this program. You must follow the exact format as sample output.

```
CP-2016-12345>              //Show Prompt if user inputed Enter
Phone Book
1. Add person
2. Remove person
3. Print phone book
CP-2016-12345>1
Select Type
1. Person
2. Work
3. Family
4. Friend
CP-2016-12345>1
Name: John doe
Phone_number: 010-1234-5678
Successfully added new person.
CP-2016-12345>
Phone Book
1. Add person
2. Remove person
3. Print phone book
CP-2016-12345>2
Enter Index of person: 10
Person does not exist! // A person is successfully deleted from the Phone Book!
CP-2016-12345>
Phone Book
1. Add person
2. Remove person
3. Print phone book
CP-2016-12345>3
Phone Book Print
1. John doe_010-1234-5678_team
2. Jane doe_02-1234-5678_age
3. Josua doe_02-8880-1234_birthday_dDay
CP-2016-12345>exit
Bye.
```

Submission – Compressed file : source code and report

Mail title: [COMP-HW1]student id_name

Compressed file name: student id_name.zip(tar)

Email : cp2016s@gmail.com.

Deadline : 04-20-2016

Your file should be named using upper case.

- **Caution**
  - **Over the deadline ; after 04-20-2016 – 20% deduction**
  - **More than 2days late - 0 point**
  - **Do not keep the file format – 20% deduction**
  - **Compile error - 0 point**
  - **Check a code copy using Clone checker – related students 0 point**
  - **You must follow good programming style which discussed in class**

- **Source Code**

  Visual, gcc file, both are acceptable. Make readme file is encouraged.

- **Report**
  - **Include specific explanation about the code**
  - **Include screen shot of the file.**
  - **PDF, DOC, HWP file.**