Submission detail:

A) Submit cpp file for each question with given main function.

B) points distribution: 50% compilation, 50% correctness.

Q1: Create a class named BankAccount. (25pt)

1. Create a class called BankAccount:
2. One private member variable - balance
3. Default constructor initialize balance to 0.0
4. One arg constructor initialize balance.
5. Create a function called deposit(double) which increase balance by arg.
6. Create a function called withdraw(double) which decrease balance by arg, balance can not be negative.
7. Create a function called displayBalance() returns the balance.
8. Accessor for balance

Use following main() to test your function.

```
int main(){
  BankAccount a(100);
  a.displayBalance();
  a.deposit(1000.75);
  a.displayBalance();
  a.withdraw(200.50);
  a.displayBalance();
  cout<<a.getBalance()<<endl;
}
```

Output from main:

```
balance: 100
balance: 1100.75
balance: 900.25
900.25
```

Answer:

Q2: Continue of previous question (25pt)

1. Create a class call SavingAccount inherits from BankAccount.
2. One private member variable - interestRate
3. Default constructor initialize - balance to 0, interestRate to 0.
4. Two arg constructor initialize - balance and interestRate.
5. Create a function addInterest() which add interest to the balance (balance += balance * interest).
6. Override withdraw(double) which decrease balance by arg and fee(20), balance can not be negative
7. Override displayBalance() returns balance of SavingAccount.

Use following main() to test your function.

```
int main(){
  SavingAccount a(19768,0.04);
  a.displayBalance();
  a.addInterest();
  a.displayBalance();
  a.deposit(7500);
  a.displayBalance();
  a.withdraw(200.50);
  a.displayBalance();
  a.withdraw(50000);
  a.displayBalance();
}
```

Output from main:

```
Saving balance: 19768
InterstRate: 0.04
Saving balance: 20558.7
InterstRate: 0.04
Saving balance: 28058.7
InterstRate: 0.04
Saving balance: 27838.2
InterstRate: 0.04
Can not withdraw, balance can not be negative.
Saving balance: 27838.2
InterstRate: 0.04
```

Answer:

Q3: Continue of previous question (25pt)

1. Convert BankAccount to abstract class.
2. Create a class call CheckingAccount inherits from BankAccount.
3. Two private member variable Fee, min_balance.
4. Default constructor initialize - balance to 100, Fee to 20, min_balance to 100.
5. Three arg constructor initialize - balance, Fee and min_balance.
6. Override withdraw(double) which decrease balance by arg, if balance is less than min_balance, then subtract fee from balance.
7. Override displayBalance() returns balance of CheckingAccount.

Use following main() to test your function.

```
int main(){
  BankAccount *a;
  a = new CheckingAccount(3000,50,1500);
  a->displayBalance();
  a->withdraw(2000);
  a->displayBalance();
  a->withdraw(2000);
  a->displayBalance();
  delete a;
  a = new SavingAccount(1500,0.01);
  a->displayBalance();
  a->withdraw(1000);
  a->displayBalance();
  a->withdraw(1000);
  a->displayBalance();
}
```

Output from main:

```
Checking balance: 3000
Checking balance: 950
Checking balance: -1100
Saving balance: 1500
InterstRate: 0.01
Saving balance: 480
InterstRate: 0.01
Can not withdraw, balance can not be negative.
Saving balance: 480
InterstRate: 0.01
```

Q4: Continue of previous question (25pt)

1. Implement a function total_balance(vector<BankAccount>) which return the total balance of all BankAccount from the given vector.

2. Implement a function max_balance(vector<BankAccount>) which return the max balance of all BankAccount from the given vector.

2. Implement a function avg_balance(vector<BankAccount>) which return the average balance of all BankAccount from the given vector.

Use following main() to test your function.

```
int main(){
  vector<BankAccount*> a;
  a.push_back(new SavingAccount(1050,0.02));
  a.push_back(new CheckingAccount(4000,50,100));
  a.push_back(new SavingAccount(2000,0.04));
  a.push_back(new CheckingAccount(3000,60,200));
  cout<<total_balance(a)<<endl;
  cout<<max_balance(a)<<endl;
  cout<<avg_balance(a)<<endl;
}
```

Output from main:

```
10050
4000
2512.5
```

Answer: