main.cpp

```cpp
// Name: Keshav Mehta, Kenry Yu
// Date: Februrary 8, 2022

#include "Account.h"
using namespace std;
int main()
{
 Account my_account(100); // Set up my account with $100
 my_account.deposit(50);
 my_account.withdraw(175); // Penalty of $20 will apply
 my_account.withdraw(25);
 cout << "Account balance: " << my_account.get_balance() << "\n";

 my_account.withdraw(my_account.get_balance()); // withdraw all
 cout << "Account balance: " << my_account.get_balance() << "\n";
 return 0;
}
```

Account.h

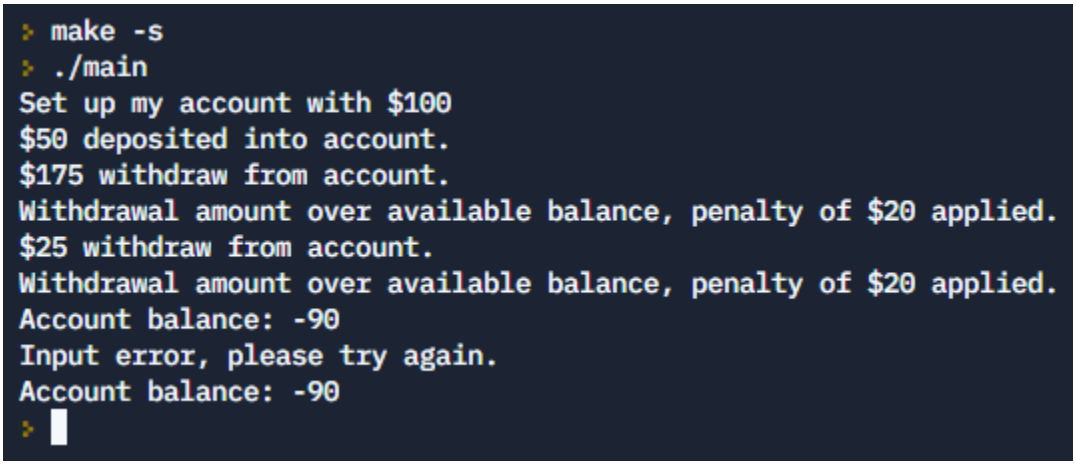
```
#ifndef ACCOUNT_H
#define ACCOUNT_H
#include <iostream>
using namespace std;

class Account
{
  // Private members
private:
  int balance;
  void penalty();

  // Public members
public:
  // Default and overload contructor
  Account();
  Account(int amount);
  // Deposit function that add amount to balance
  void deposit(int amount);
  // Withdraw function that take money out from balance
  void withdraw(int amount);
  // get_balance function that return balance
  int get_balance();
  // display function that output statement with balance
  void display();
};

#endif
```

Account.cpp

```cpp
#include "Account.h"
#include <iostream>
using namespace std;
// Default constructor: sets up an account without an initial balance
Account::Account() {
  this->balance = 0;
  cout << "Set up my account with $" << get_balance() << endl;
}
// Overload constructor: sets up an account with a specific initial balance
Account::Account(int amount) {
  this->balance = amount;
  cout << "Set up my account with $" << get_balance() << endl;
}
// Deposit function: deposits a valid non-negative ammount
void Account::deposit(int amount) {
  if (amount < 0)
    cout << "Input error, please try again.\n";
  else {
    cout << "$" << amount << " deposited into account.\n";
    this->balance += amount;
  }
}
// Withdraw function: withdraw a valid positive ammount
void Account::withdraw(int amount) {
  if (amount < 0)
    cout << "Input error, please try again.\n";
  else {
    this->balance -= amount;
    cout << "$" << amount << " withdraw from account.\n";
    if (balance < 0) {
      penalty();
    }
  }
}
// GetBalance function: returns the current balance of the account
int Account::get_balance() { return this->balance; }
// Penalty function: penalise if the balance is lower than the withdraw ammount
void Account::penalty() {
  this->balance -= 20;
  cout << "Withdrawal amount over available balance, penalty of $20 applied."
       << endl;
}
// Display function: displays the account status
```

```
void Account::display() {
  cout << "Current available balance in account is $" << get_balance() << endl;
}
```

Output


make -s
./main
Set up my account with $100
$50 deposited into account.
$175 withdraw from account.
Withdrawal amount over available balance, penalty of $20 applied.
$25 withdraw from account.
Withdrawal amount over available balance, penalty of $20 applied.
Account balance: -90
Input error, please try again.
Account balance: -90