Faysal Khatri
CSC220 -- Activity 9
2017-07-3

Sample Output

```
Employee 3:
    Name: John Franklin
 Address: 222 Bank
   Phone: 480-555-2343
Current pay : 3000

Employee 4:
    Name: Faysal Contractor1
 Address: 222 Contractor
   Phone: 480-555-9999
Current pay : 2000

Employee 5:
    Name: Faysal Contractor2
 Address: 222 Contractor
   Phone: 480-555-9999
Exceeded max hours. Only max hours will be recorded
Current pay : 2000

Employee 6:
    Name: Faysal Contractor3
 Address: 222 Contractor
   Phone: 480-555-9999
Current pay : 4700

Employee 7:
    Name: Faysal Contractor4
 Address: 222 Contractor
   Phone: 480-555-9999
Exceeded max hours. Only max hours will be recorded
Current pay : 3500

Total payroll: 17523
Done!
```

contractor.hpp

```
// Staff member - base class

#include<iostream>
#include "hourly.hpp"

class Contractor : public Hourly {
  private:
       int maxHours = 40;

  public:
    Contractor(std::string n, std::string a, std::string p, std::string s, int r, int h,
int m) : Hourly(n, a, p, s, r, h)
    {
      maxHours = m;
    }

      Contractor(std::string n, std::string a, std::string p, std::string s, int r, int
h) : Hourly(n, a, p, s, r, h)
    {
      maxHours = 40;
    }

  int pay();
 void setMaxHours(int m);
};
```

contractor.cpp

```


#include "contractor.hpp"

int Contractor::pay() {
  int pay = 0;

  if (hours > maxHours) {
        std::cout << "Exceeded max hours. Only max hours will be recorded\n";
        hours = maxHours;
  }

  if ( hours > 40 ) {
        pay = payrate * 40;
        pay += (payrate * 1.5) * (hours - 40);
  }
  else {
        pay = payrate * hours;
  }

  return (pay);
}


void Contractor::setMaxHours(int m) {
      maxHours = m;
}
```

staff.cpp

```
// Adapted from an example from Java Foundations
//  by John Lewis, Peter DePasquale, & Joe Chase
//
#include<iostream>
#include"staffmember.hpp"
#include"salaried.hpp"
#include"volunteer.hpp"
#include"hourly.hpp"
#include"contractor.hpp"

using namespace std;

#define STAFF_SIZE  8

void print_paystub(StaffMember * s) {
  s->print();
  cout << "Current pay : " << s->pay() << endl;
}

int main() {
  int ii;
  int totalpay = 0;
  Salaried *  emp1 = new Salaried("Bob Smith", "123 Main", "480-555-8765",
                          "111-22-3333", 50000);
  Volunteer * emp2 =
          new Volunteer("Alice Jenkins", "456 4th", "480-555-2341");
  Hourly * emp3 =
          new Hourly("Mitch Jenkins", "456 4th", "480-555-2341",
                          "222-33-2222", 10, 40);

  StaffMember * staff[STAFF_SIZE];
  staff[0] = emp1;
  staff[1] = emp2;
  staff[2] = emp3;
  staff[3] = new Salaried("John Franklin", "222 Bank", "480-555-2343", "111-22-4444",
78000);
  staff[4] = new Contractor("Faysal Contractor1", "222 Contractor", "480-555-9999", "111-
22-1111", 50, 40);
  staff[5] = new Contractor("Faysal Contractor2", "222 Contractor", "480-555-9999", "111-
22-1111", 50, 70);
  staff[6] = new Contractor("Faysal Contractor3", "222 Contractor", "480-555-9999", "111-
22-1111", 50, 76, 80);
  staff[7] = new Contractor("Faysal Contractor4", "222 Contractor", "480-555-9999", "111-
22-1111", 50, 90, 60);

  for(ii=0;ii<STAFF_SIZE;ii++)  {
    cout <<"\nEmployee " << ii << ": " << endl;
    print_paystub(staff[ii]);
    totalpay=totalpay + staff[ii]->pay();
  }

  cout << "\nTotal payroll: " << totalpay << endl;
  cout << "Done!" << endl;

  return 0;

}
```