```cpp
#include <iostream>
#include <string>

class Employee {
public:
    std::string name;
    int age;
    float hoursWorked;
    float hourlyRate;

    virtual float calculatePay() = 0; // abstract method
};

class FullTimeEmployee : public Employee {
public:
    FullTimeEmployee() {
        hourlyRate = 500.0;
    }

    float calculatePay() {
        float overtimeHours = hoursWorked > 40 ? hoursWorked - 40 : 0;
        float regularHours = hoursWorked - overtimeHours;

        return regularHours * hourlyRate + overtimeHours * 2 * hourlyRate;
    }
};

class PartTimeEmployee : public Employee {
public:
    PartTimeEmployee() {
        hourlyRate = 750.0;
    }

    float calculatePay() {
        float overtimeHours = hoursWorked > 20 ? hoursWorked - 20 : 0;
        float regularHours = hoursWorked - overtimeHours;

        return regularHours * hourlyRate + overtimeHours * 1.5 * hourlyRate;
    }
};

int main() {
    std::string empType;
    std::cout << "Enter employee type (full-time/part-time): ";
    std::cin >> empType;

    Employee *emp;
    if (empType == "full-time") {
        emp = new FullTimeEmployee();
    } else {
        emp = new PartTimeEmployee();
    }

    std::cout << "Enter number of hours worked: ";
    std::cin >> emp->hoursWorked;

    float totalPay = emp->calculatePay();

    std::cout << "Total pay: " << totalPay << std::endl;

    delete emp;
    return 0;
}
```