

```

1  #include <iostream>
2  #include <string>
3
4  class Employee {
5  public:
6      std::string name;
7      int age;
8      float hoursWorked;
9      float hourlyRate;
10
11     virtual float calculatePay() = 0; // abstract method
12 };
13
14 class FullTimeEmployee : public Employee {
15 public:
16     FullTimeEmployee() {
17         hourlyRate = 500.0;
18     }
19
20     float calculatePay() {
21         float overtimeHours = hoursWorked > 40 ? hoursWorked - 40 : 0;
22         float regularHours = hoursWorked - overtimeHours;
23
24         return regularHours * hourlyRate + overtimeHours * 2 * hourlyRate;
25     }
26 };
27
28 class PartTimeEmployee : public Employee {
29 public:
30     PartTimeEmployee() {
31         hourlyRate = 750.0;
32     }
33
34     float calculatePay() {
35         float overtimeHours = hoursWorked > 20 ? hoursWorked - 20 : 0;
36         float regularHours = hoursWorked - overtimeHours;
37
38         return regularHours * hourlyRate + overtimeHours * 1.5 * hourlyRate;
39     }
40 };
41
42 int main() {
43     std::string empType;
44     std::cout << "Enter employee type (full-time/part-time): ";
45     std::cin >> empType;
46
47     Employee *emp;
48     if (empType == "full-time") {
49         emp = new FullTimeEmployee();
50     } else {
51         emp = new PartTimeEmployee();
52     }
53
54     std::cout << "Enter number of hours worked: ";
55     std::cin >> emp->hoursWorked;
56
57     float totalPay = emp->calculatePay();
58
59     std::cout << "Total pay: " << totalPay << std::endl;
60
61     delete emp;
62     return 0;
63 }

```