```c
// Glenn Hewey
// Cecs 424
// Lab 3
// March 23 2018
#include <stdio.h>
#include <stdlib.h>

struct Employee {
  void** vtable;
  int age;
};

struct HourlyEmployee {
  void** vtable;
  int age;
  double hourly_rate;
  double hours;
};

struct CommissionEmployee {
  void** vtable;
  int age;
  double sales_amount;
};

struct SeniorSalesman {
  void** vtable;
  int age;
  double sales_amount;
};

// Function declaration
void Speak_Hourly(struct Employee* employee);
void Speak_Commission(struct Employee* employee);
void Speak_SeniorSalesman(struct Employee* employee);

double GetPay_Hourly(struct Employee* employee);
double GetPay_Commission(struct Employee* employee);
double GetPay_SeniorSalesman(struct Employee* employee);

void Construct_Hourly(struct HourlyEmployee* hEmployee);
void Construct_Commission(struct CommissionEmployee* cEmployee);
void Construct_SeniorSalesman(struct SeniorSalesman* sEmployee);

// vtables for hourly, commission, and senior saleman
void* Vtable_Hourly[2] = {Speak_Hourly, GetPay_Hourly};
void* Vtable_Commission[2] = {Speak_Commission, GetPay_Commission};
void* Vtable_SeniorSalesman[2] = {Speak_Commission, GetPay_SeniorSalesman};

// Outputs the speak method for hourly employees
void Speak_Hourly(struct Employee* employee){
  printf("I work for %.2lf dollars per hour.\n", ((struct HourlyEmployee *)
employee)->hourly_rate);
}

// Outputs the speak method for commission and senior salesman employees
void Speak_Commission(struct Employee* employee){
  printf("I make commission on %.2lf dollars in sales!\n", ((struct
CommissionEmployee *) employee)->sales_amount);
}
```

```c

// Returns a double of how much the hourly employee makes
double GetPay_Hourly(struct Employee* employee){
  return ((struct HourlyEmployee *) employee)->hourly_rate * ((struct HourlyEmployee *) employee)->hours;
}

// Returns a double of how much the commission employee makes
double GetPay_Commission(struct Employee* employee){
  return (0.1*((struct CommissionEmployee *) employee)->sales_amount) + 40000.0;
}

// Returns a double of how much the senior salesman makes
double GetPay_SeniorSalesman(struct Employee* employee){
  double percent = 0.2;
  if( ((struct SeniorSalesman *) employee)->age >= 40 ){
    percent += 0.05;
  }
  return (percent*((struct SeniorSalesman *) employee)->sales_amount) + 50000.0;
}

// Constructs an hourly employee initializes values to 0 and points to the corrent vtable
void Construct_Hourly(struct HourlyEmployee *hEmployee){
  hEmployee->vtable = Vtable_Hourly;
  hEmployee->age = 0;
  hEmployee->hourly_rate = 0;
  hEmployee->hours = 0;
}

// Constructs a commission employee initializes values to 0 and points to the corrent vtable
void Construct_Commission(struct CommissionEmployee *cEmployee){
  cEmployee->vtable = Vtable_Commission;
  cEmployee->age = 0;
  cEmployee->sales_amount = 0;
}

// Constructs a senior salesman initializes values to 0 and points to the corrent vtable
void Construct_SeniorSalesman(struct SeniorSalesman *sEmployee){
  sEmployee->vtable = Vtable_SeniorSalesman;
  sEmployee->age = 0;
  sEmployee->sales_amount = 0;
}

int main()
{
  // declare employee pointer variable
  struct Employee* e;

  // user input to choose what type of employee
  int i;
  printf("input employee type\n1) Hourly \n2) Commission \n3) Senior Salesman \n");
  scanf("%d", &i);

  switch(i) {
    case 1:
      printf("You chose Hourly\n");

      // allocate using malloc
      e = malloc(sizeof(struct HourlyEmployee));

      // Use constructor
      Construct_Hourly((struct HourlyEmployee *)e);

      // Ask the user how old the employee is
      printf("input age: ");
      scanf("%d", &(e)->age);

      // Ask user for rate
      printf("input rate: ");
      scanf("%lf", &((struct HourlyEmployee *)e)->hourly_rate);

      // Ask user for hours
      printf("input hours: ");
      scanf("%lf", &((struct HourlyEmployee *)e)->hours);

      break;
    case 2:
      printf("You chose Commission\n");

      // allocate using malloc
      e = malloc(sizeof(struct CommissionEmployee));

      // Use constructor
      Construct_Commission((struct CommissionEmployee *)e);

      // Ask the user how old the employee is
      printf("input age: ");
      scanf("%d", &(e)->age);

      // Ask user for sales
      printf("input sales: ");
      scanf("%lf", &((struct CommissionEmployee *)e)->sales_amount);

      break;
    case 3:
      printf("You chose Senior Salesman\n");

      // allocate using malloc
      e = malloc(sizeof(struct SeniorSalesman));

      // Use constructor
      Construct_SeniorSalesman((struct SeniorSalesman *)e);

      // Ask the user how old the employee is
      printf("input age: ");
      scanf("%d", &(e)->age);

      // Ask user for sales
      printf("input sales: ");
      scanf("%lf", &((struct SeniorSalesman *)e)->sales_amount);

      break;
    default:
      break;
```

```c
  }

  // Invoke speak method using employee vtable
  ((void (*) (struct Employee*))e->vtable[0])((struct Employee *)e);

  // Invoke getPay method using employee vtable
  printf("I make: %.2lf\n", ((double (*) (struct Employee*))e->vtable[1])
((struct Employee *)e));

  free(e);

  return 0;
}
```