Solution Review: Check If the Person Has a Driving License

This lesson gives a detailed solution review to the challenge in the previous lesson.

We'll cover the following

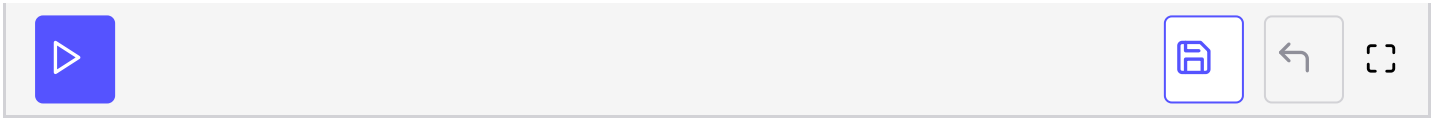
- Solution:
  - Explanation

Solution:

```
#![allow(dead_code)]
//declare a structure
struct Car {
   owner_age:i32,
}
struct Motorbike {
   owner_age:i32,
}
//declare a trait
trait Drive {
   fn can_drive(&self)->i32;
}
//implement the trait
impl Drive for Car{
   fn can_drive(&self)->i32{
      if self.owner_age>18 {
         1
      }
      else {
         0
      }
   }
}
//implement the trait
impl Drive for Motorbike{
   fn can_drive(&self)->i32{
      if self.owner_age>14{
         1
      }
      else {
         0
      }
   }
}
fn main(){
   let mut c = Car {
```

```
      owner_age:16
   };
   println!("Can age = 16 drive a car ? - {}", c.can_drive());

   c.owner_age = 23;
   println!("Can age = 23 drive a car ? - {}", c.can_drive());
   let mut d = Motorbike  {
      owner_age:10
   };
   println!("Can age = 10 drive a motorbike? - {}", d.can_drive());
   d.owner_age = 17;
   println!("Can age = 17 drive a motorbike? - {}", d.can_drive());
}
```



Explanation

- `struct` `Car`
  - A `struct` `Car` is declared from **line 3 to line 5** which has item `owner_age` type `i32`
- `struct` `Motorbike`
  - A `struct` `Motorbike` is declared from **line 6 to line 8** which has item `owner_age` type `i32`
- `trait` `Drive`
  - A `trait` `Drive` is declared on **line 10**. It has an abstract method `can_drive`.
- `impl` `Drive` `for` `Car`
  - A function `can_drive` is defined from **line 14 to line 23** within the trait which takes a parameter `&self` and returns an integer value.
    - Within the function, an `if..else` construct is used. If the `age` is greater than 18, the function returns 1 else it returns 0.
- `impl` `Drive` `for` `Motorbike`
  - A function `can_drive` is defined from **line 35 to line 34** within the trait which takes a parameter `&self` and returns an integer value.
    - Within the function, an `if..else` statement is used. If the `age` is greater than 14, the function returns 1 else it returns 0.

Now you have learned about traits and generics, let’s learn to organize code using “Modules” in the next chapter.