

ECE326 – Fall 2019: Week 13 Exercise Questions

1. True or False [1 mark each]

Circle T is true, otherwise circle F for false.

1. Trait objects can only be created on the heap. **T** **F**
2. Both C++ and Rust allows specifying the return type of arithmetic operations. **T** **F**
3. Generally speaking, a chain of iterator adapters should end in collect or fold. **T** **F**
4. In C++, closures and functions can be used interchangeably. **T** **F**
5. In a multithreaded setting, mutually recursive functions must implement mutual exclusion.
 T **F**
6. A generator cannot have the return statement. **T** **F**

2. Average of Shapes [10 marks]

Implement the Shape trait for Rectangle, Circle, and Triangle. The Shape trait has one method, area, which returns the area of the shape. Write a function which takes a list of shapes and returns the average area.

```
fn average_area(list: &Vec<Box<dyn Shape>>) -> f64;
```

3. Coroutine [10 marks]

Write a Python function, ppmt, which returns the payment on principal for a given period for a loan based on periodic, constant payments and a constant interest rate. In addition, at any time, the user can make either lump sum payment or take out additional loan, which would cause the next payment to change based on remaining period.

```
# rate: interest rate (e.g. 0.0375 for 3.75% per period)
# per: number of periods
# pv: initial principal value on loan
def ppmt(rate, per, pv)

loan = ppmt(0.0375, 240, 1000000)
first_payment = next(loan)
# e.g. taking out additional loan
second_payment = loan.send(50000)
```