

ECE326 – Fall 2019: Week 11 Exercise Questions

1. True or False [1 mark each]

Circle T is true, otherwise circle F for false.

1. In Rust, literals are first class citizens. **T** **F**
2. Box forces the contained object to be heap allocated. **T** **F**
3. A mutable object can be modified even if an immutable reference of it is in scope. **T** **F**
4. After transferring ownership, an immutable object can become mutable. **T** **F**
5. Generics in Rust supports duck typing. **T** **F**

2. Match [10 marks]

Create a function named `hand_score` which takes a tuple of two characters and returns the score of a blackjack hand using ONLY the match expression. You need to handle soft hands (e.g. A8 = 19) and hard hands. Return 21 for Blackjack. Assume the card values have been sanitized (do not check for invalid card). Hint: `is_digit()` and `to_digit()` may be helpful.

<https://doc.rust-lang.org/std/primitive.char.html>

```
fn hand_score(hand: (char, char)) -> u32
```

3. Structure and Method [20 marks]

Write structure named `Matrix` which supports a 2x2 matrix of type `f64`, and implements the determinant method, the transpose method, and the inverse method. The transpose method should modify the existing matrix, but the inverse method should return a new matrix if the matrix is invertible, otherwise it should return `None` (Hint: use `Option<T>`). Write a new static method which creates and initializes the matrix. Implement the `Display` trait for `Matrix`.

4. Generic Function [10 marks]

Write a generic function, `sum_of_squares`, which calculates the sum of squares of a generic array slice. Hint: you may need some trait bounds.