1. Implement an enum to represent the suite of a playing card

enum Suite {

Diamonds,

Hearts,

Spades,

Clubs

}

1. Instantiate ‘hearts’ instance of the enum created above

let hearts = Suite::Hearts;

1. Implement an enum to represent the face value of a card. Enum variants should include face, number, ace, and joker. Include appropriate associated values for each variant to distinguish between card values in the same variant.

enum Value {

Face(String),

Number(i32),

Ace,

Joker

}

1. Instantiate a Joker instance, an ace, a five, and a king in separate constants

let joker = Value::Joker;

let ace = Value::Ace;

let five = Value::Number(5);

let king = Value::Face(String::from(“King”));

1. Explain the advantage of Option over null references

Option forces the programmer to explicitly denote and handle when a value is possibly null. Instead, with null references programmers often accidently assume a value cannot be null, causing an invalid null reference.