Exercises 2 (Values and types)

2A. *(Primitive types)*

Consider an application that processes amounts of money up to £100,000.00. Which primitive type would you use to represent such amounts: (a) in C; (b) in Java?

2B. *(Composite types)*

Answer this question using the notation of cartesian products, disjoint unions, and mappings.

(a) Write down the set of values of each of the C types defined below, and the cardinality of each type:

```
enum Suit {CLUB, DIAMOND, HEART, SPADE};
struct Card {Suit suit; byte rank;};
typedef Card[] Hand;
enum Option {PASS, PLAY};
struct Turn {Option opt; Card card;};
```

(b) Write down the set of objects in a Java program that includes the following class definitions:

```
class A { int i; float f; }
class B { boolean b; }
class C extends B { char c; }
```

(c) Suppose now that `B` is changed to an abstract class. Modify your answer accordingly.

2C. *(Relationship between arrays and functions)*

(a) Implement the mapping {*false* → *true*, *true* → *false*}, using (i) an array and (ii) a function, in your favorite imperative programming language.

(b) Implement the factorial function over the integers 0 through 10, using (i) an array and (ii) a function, in your favorite imperative programming language.

(c) In what ways are arrays and functions fundamentally different? Answer this in terms of the *essential* properties of arrays and functions, neglecting any peculiarities that arrays or functions might have in your favorite language.

2D. *(Type systems)*

Choose a favorite programming language (other than Java). Systematically analyze your language's type system, in the same way as various languages have been analyzed in §2 of the course notes.

(a) What primitive types does your language support?

(b) What composite types does your language support? Express the set of values of each composite type using the notation of cartesian products, disjoint unions, and mappings.

(c) Can recursive types be defined in your language? If so, how?

(d) Is your language statically or dynamically typed?

2E. *(Static vs dynamic typing)*

(a) Find a program you have written in a *statically* typed language that would have been simpler to write in a dynamically typed language.

(b) Find a program you have written in a *dynamically* typed language that could equally well have been written in a statically typed language.