COMP 141: Data Types— Part 1

Instructions: In this exercise, we are going to study different preliminary concepts on data types.

(1) Consider the following class definition in Java.

```
public class CC {

    int i = 0;

    public int f (int j) {
        if (i) return i; else return 0;
    }
}
```

When we try to compile it, we get the following error message:

```
javac CC.java

CC.java:8: error: incompatible types: int cannot be converted to boolean
        if (i) return i; else return 0;
            ^
1 error
```

Based on this demo, identify whether Java

(a) is explicitly-typed or implicitly typed? Why?

(b) is statically-typed or dynamically-typed? Why?

(c) is strongly-typed or weakly-typed? Why?

(d) does type inference or type checking? Why?

(2) Consider the following program in C.

```
#include <stdio.h>

enum Day {Monday=-5, Tuesday, Wednesday};

int main () {
        enum Day x = Tuesday;
        printf("%d\n",++x);
        return 0;
}
```

What would be the output? Why?

(3) Define the enumerated type `Direction` in Haskell consisting of values: `Up`, `Down`, `Left`, `Right`.

(4) Consider the following program in C.

```
#include <stdio.h>


struct IntChar {
        int i;
        char c;
};
```

```
int main () {
        struct IntChar y;
        y.i = 0;
        y.c = 'a';
        printf("%c\n",y.c);
        printf("%d\n",y.i);
        return 0;
}
```

What would be the output?

(5) Define a value in Haskell whose type is the Cartesian product of a boolean, a string, and a character.

(6) Consider the following program in C.

```
#include <stdio.h>

union IntOrChar {
        int i;
        char c;
};


int main () {
        union IntOrChar y;
        y.i = 0;
        y.c = 'a';
        printf("%c\n",y.c);
        printf("%d\n",y.i);
        return 0;
}
```

What would be the output? Compare it with the previous program and explain why (You can assume that integers take 2 bytes).

(7) Consider the following Haskell program.

```
ghci> maximum x y = if x + y then x else y

<interactive>:2:24:
    No instance for (Num Bool) arising from a use of `+'
    In the expression: x + y
    In the expression: if x + y then x else y
    In an equation for `maximum': maximum x y = if x + y then x else y
```

According to this, identify whether

(a) this program is explicitly typed or implicitly typed? Why?

(b) Haskell is statically-typed or dynamically-typed? Why?

(c) Haskell is strongly-typed or weakly-typed? Why?

(d) Haskell translator has done type inference or type checking? Why?

(8) Define a C program in which

- a union of integers, strings and floating point numbers is defined.

- a variable of this union type is defined.
- In order, first string “aaaa” is assigned to this variable. Next, floating point number 834.63 is assigned to it. And finally, integer 47 is assigned to it.
- After the assignments, print the content that this variable holds first as an integer, next as a string, and finally as a floating point number.

Submit your C program and report the output of this program.