

1. (a) Explain the following programming paradigms:

i. Imperative [5 points]

Control flow is an explicit sequence of commands.

ii. Procedural [5 points]

Imperative Programming organized around hierarchies of nested procedure cells.

iii. Functional [5 points]

Computation proceeds by (nested) function calls that avoid any global state mutation and through the definition of function composition.

(b) How does the procedural paradigm improve over the imperative paradigm? [5 points]

by adding layers of abstraction in the form procedures, (organizing code into procedures which allows for code reusability and modularity).

procedures interact through well-defined contracts and can encapsulate local variables. It's testable, this makes the code easier to read, understand and maintain.

(c) How does the functional paradigm improve over the procedural paradigm? [5 points]

by encourages the usage of pure functions (with no share state and mutation) which make testing, formal verification and concurrency easier.

Question 2 :

```
type Product = {  
    name: string;  
    price: number;  
    discounted: boolean;  
}
```

```
const getDiscountedProductAveragePrice = (inventory: Product[]): number => {  
  
    let discountedProducts = inventory.filter(Product => Product.discount);  
  
    let discountedPriceSum = discountedProducts.map(Product => Product.price)  
        .reduce((sum, price) => sum + price, 0);  
  
    return discountedProducts.length > 0 ? discountedPriceSum / discountedProducts.length : 0;  
}
```

3. Write the most specific types for the following expressions. Guidelines: arrays must be homogeneous, arithmetic operations must be performed on numbers, use generics where possible, and avoid using any.

(a) $(x, y) \Rightarrow x.\text{some}(y)$ [5 points]

$\langle T \rangle (x: T[], y: (z: T) \Rightarrow \text{boolean}) \Rightarrow \text{boolean}$

(b) $x \Rightarrow x.\text{reduce}((\text{acc}, \text{cur}) \Rightarrow \text{acc} + \text{cur}, 0)$ [5 points]

$(x: \text{number}[]) \Rightarrow \text{number}$

(c) $(x, y) \Rightarrow x ? y[0] : y[1]$ [5 points]

$\langle T \rangle (x: \text{boolean}, y: T[]) \Rightarrow T$

(d) $(f, g) \Rightarrow x \Rightarrow f(g(x+1))$ [5 points]

$\langle T_1, T_2 \rangle (f: (v: T_1) \Rightarrow T_2, g: (a: \text{number}) \Rightarrow T_1) \Rightarrow (x: \text{number}) \Rightarrow T_2$