**Objects and Classes (overview)**

* Object (definition): is an instance of a particular class or subclass with the class’s own methods or procedures and data variables.
* Class (definition): a template/blueprint for creating different objects which is defined its behaviour/state (properties).
* Example of classes: have three different main classes: Company, Consumer, Plan
* Example of object: the plan list

**Properties and methods:**

* Properties (definition): used to maintain a list of values in which the key is a string and the value is also a string.
* Methods (definition): is a collection of statements in the code that are grouped together to perform an operation that could return something back when needed (when call function is used)
* Example 1: in company class: addConsumer (Consumer [] comsumerList) adds a consumer in the consumer list.
* Example 2: in consumer class: (String name, Company [] companyList, Plan [] planList) takes a name, a list of companies, and a list of plans

**Inheritance:**

* (definition): when a child class takes the same attributes and behaviours as the parent class but have different attributes and behaviours then more than one child class (from each other)
* Example: parent class: plan class, child class: Data plan, Long Distance plan, Talk plan.
* Talk plan offers unlimited talk and text that only exist in the talk plan not the data plan, also only long distance plan will only offer long distance not in the Talk plan. And lastly only the data plan will offer data not long distance plan.

**Overloading:**

* (definition): a method (function) that has the same name but all have different number of parameters (or arguments)
* Example:

Company () – company list

Company (String name) – takes a name only

Company (String name, Plan [] plans) - takes a name, adds plans plan list

Company (String name, Plan [] plans, Consumer [] consumer) - takes a name, adds plans plan list and adds consumer to consumer list.

**Overriding:**

* (definition): allows child class to provide a specific implementation of a method that is already provided by the parent class.
* Example: getMonthlyCharges (): float – appears as a constructor under Plan (PARENT) class AND its CHILD classes.
* The child classes (DataPlan, LongDistancePlan, and TalkPlan) overrides the attributes that were given to them by the parent class (Plan) -> like the amount of cost for DataPlan is going to be different from the amount of cost for LongDistancePlan and TalkPlan monthly.

**Polymorphism**

* Definition: when a parent class is used to refer to a child class object.
* Example: in the main program, we reference the child class object “DataPlan” in the Parent Class Plan list.

**NEW Introduction:**

We understand that the cellphone plans are getting out of control, and companies must struggle to keep a track of everything from its clients, their plans and the company’s monthly income. But what if there was something that could do these things? Then wait no more, we introduce CellTech, our company manages AND tracks a set of clients and plans that include text rates, data rates, talk minutes, etc, that the consumers will have chosen after choosing companies, like Trump Network, Cringe INC, and United Men Limited that provides the consumer with a plan that is best suited to their lifestyle.

**Conclusion:**

By choosing CellTech, companies will be able to keep a track of their client and their desired plan as well as receive monthly income from their consumers. Our program is designed to help you remain organized and stay ahead of other competitive cellphone companies. We hope that you will give us the opportunity to work with you in the near future. Thank you for listening.