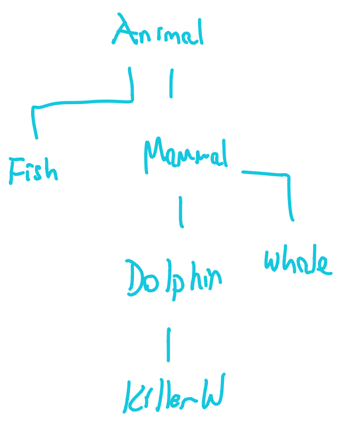
Quiz 3 – CS 315

Name: ____________________, ID: ____________________

1) Consider the following Java code:

```
abstract class Animal {
        public abstract void speak();
}
class Fish extends Animal {
        public void speak() {}
}
class Mammal extends Animal {
        public void speak() {}
}
class Dolphin extends Mammal {
        public void speak() {}
}
class KillerWhale extends Dolphin {
        public void speak() {}
}
class Whale extends Mammal {
        public void speak() {}
}
```



Write the type of the conversion (one of *narrowing* (*N*), *widening* (*W*), *invalid* (*I*)) for each and whether the code *compiles* (y/n) and *runs* (y/n). Assume lines are executed in succession. If a line has a compile error, assume it is not executed. A narrowing conversion is one that converts a base type to a derived type. A widening conversion is one that converts a derived type to a base type.

```
Animal a = new Dolphin();          // conversion: _W_, compiles: _Y_, runs: _Y_
Whale w = new KillerWhale();       // conversion: _I_, compiles: _N_, runs: _N_
Dolphin d = (Dolphin) a;           // conversion: _Y_, compiles: _Y_, runs: _Y_
Mammal m = d;                      // conversion: _W_, compiles: _Y_, runs: _Y_
Fish f = (Fish) m;                 // conversion: _I_, compiles: _N_, runs: _N_
Dolphin k = (KillerWhale) d;       // conversion: _I_/_N_, compiles: _Y_, runs: _N_
```

2) You are designing a programming language and are working on defining the string type (or types) for it. Your design is expected to satisfy the following requirements:

- In order to support performance critical applications, you are required to support strings that have low memory overhead and fast indexing for the common case of ASCII-compatible text, yet you want to be able to hold Unicode text as well. For these applications, you want to be able to quickly look into the string data, but are not required to operate on Unicode characters.
- In order to support internationalization and applications that interact with human users in their native language, you are also required to support strings that can hold Unicode text and provide logical character access and string operators that work based on logical character indexes. These applications are not performance sensitive.

Design string-related types for your language so that the above requirements are met. You can design more than one string type or a single string type. Same for the character types. For your string type(s), specify the operations they support and the internal encoding they use (if there is one). For character types, define their size and what they represent. Discuss trade-offs and contrast your solution with alternative designs.

Use the backside of the paper for your answer.