

CISC 3115

# Functional Interface and Lambda Expression

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

# Outline

- Recap
  - Inheritance and polymorphism
  - Abstract method and class
- Functional Interface
- Lambda Expression

# Anonymous Class

- Essentially, a local class without a name
- Created by declaring and instantiating a class at the same time
- Use it when need a local class only once

```
class OuterClass {  
    ...  
  
    { ...  
        ParentClass a = new ParentClass() { ...}  
    }  
    ...  
}
```

# Functional Interface

- Any interface that contains only one abstract method
  - Since Java 8, a functional interface may contain one or more default methods or static methods

# Use Functional Interface

- In your own design, sometime functional interface is better choice
- More often, you use functional interfaces because some Java API methods require them
  - Examples:
    - <https://docs.oracle.com/javase/8/docs/api/java/util/function/package-summary.html>

# Functional Interface and Anonymous Class


- You can declare and instantiate a local class, or more often an anonymous class
- Example

```
ArrayList<Person> personList = new ArrayList<Person>();  
Arrays.sort(personList, new Comparator<Person> {  
    @Override  
    public int compare(Person lhs, Person rhs) {  
        // buggy (what if rhs is null?)  
        return lhs.getName().compareTo(rhs.getName());  
    }  
}
```

# Lambda Expression

- A simple way to declare and instantiate a class

```
ArrayList<Person> personList = new ArrayList<Person>();  
Arrays.sort(personList, new Comparator<Person> {  
    @Override  
    public int compare(Person lhs, Person rhs) {  
        // buggy (what if rhs is null?)  
        return lhs.getName().compareTo(rhs.getName());  
    }  
}
```



```
ArrayList<Person> personList = new ArrayList<Person>();  
Arrays.sort(personList, (lhs, rhs) -> lhs.getName().compareTo(rhs.getName()))}
```

# Questions

- Functional interface
- Lambda expression
- We shall discuss this topic again
  - Local class
  - Anonymous class
  - Static class
  - ...