Video Name: Polymorphism

Topics:

• polymorphism

Java Class(es): BankingAccount, SavingsAccount, Bank

Polymorphism: many forms

Ad hoc polymorphism uses overloaded operator (+)

Subtyping polymorphism relates to inheritance

Child class can be used in place of parent class (because child “is-a” parent). Common usage: collection of parent class which contains objects of both parent and child class.

```java
BankingAccount [] accounts = new BankingAccount[MAX_ACCOUNTS];
accounts[0] = new BankingAccount(2500, "Sally");
accounts[1] = new SavingsAccount(7500, "Morgan", 0.02);
```

C++ requires virtual keyword to ensure child class methods are called (when they override the parent class method). This allows programmer to choose early binding (compile time) or late binding (runtime). Early binding (compile time) is faster, because if binding is done at run time, each function call must first determine which function to call (i.e., parent function or child function).

In Java, binding is always done at runtime – so no need for virtual keyword.

This code will call the parent toString() method for accounts[0] and the child toString() method for accounts[1].

```java
for (int i=0; i<2; i++)
    System.out.println(accounts[i]);
```

It is not possible to assign a parent object to a child variable – this is ILLEGAL:

```java
//SavingsAccount savings = new BankingAccount(3000, "Tim", 0.01);
```

It is not possible to call child class functions using a parent class variable.

```java
BankingAccount banking = new SavingsAccount(3000, "Tim", 0.01);
// Even though banking contains SavingsAccount, can’t call
//banking.addInterest();
```
Software engineering. Can use instanceof operator, but that would tend to lead to fragile code. Better approach: add generic method for parent class that could be called for any type of account. Typically best to make this method abstract (to avoid violations of Liskov Substitution Principle, covered later)