Late Binding and Polymorphism

pm_jat@daiict.ac.in

Java reference variables are polymorphic variables

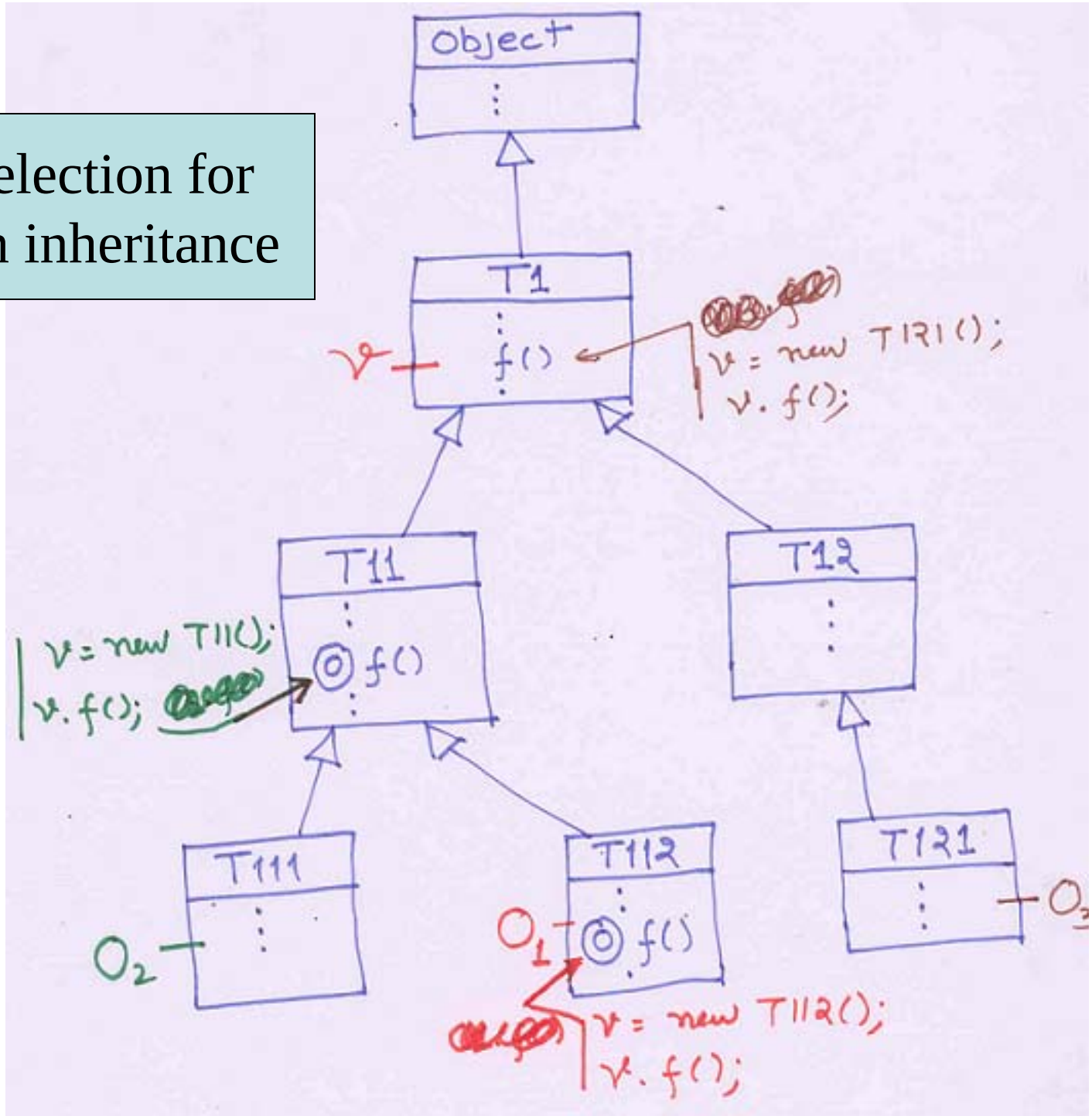
- Polymorphic variable is variable that can reference more than one type of object during run-time.

- In Java, reference variables are polymorphic, and
  - can point to any of its subtype objects
  - embodies type substitution in Java

Type Checking in Java

- In Java, *polymorphic* variables have two types- Static Type and Dynamic Type.
- Type declared is its static type; and type of object to which a variable point is its dynamic type.
- Dynamic type of a variable can change during the program execution.
- Method binding is done based on dynamic type of the variable.

Method selection for binding in inheritance



Method selection for binding in inheritance

- Consider inheritance hierarchy on next slide.
- When a variable v of type T1 refers to an object of type T112, then binding of message f to v would be bound to method f of T112
- When v refers to an object of type T111, then binding of message then binding of message f to v would be bound to method f of T11, as T111 does not override it.
- When v refers to an object of type T121, then binding of then binding of message f to v would be bound to method f of T1; because nearest class (in parent hierarchy that implement this method is f(); neither T121, nor T12 overrides it.

Where do you use polymorphic variables

- You rarely use polymorphic variables as following

```
BankAccount b = new SavingAccount();
```

- Normally polymorphic variables are used -
  - Parameter to methods
  - Having collection of objects

Parameter to Methods

- To reuse the method code, it is implemented for some super type and objects of sub-classes are passed, and object specific behavior is accomplished.

- Examples, we have seen
  - Measurable in Add method of DataSet
  - Transfer method in BankAccount class
  - Object in println methods
  - And many like this.

Actual object referred by x parameter would be any of its subtype, and message **`getMeasure`** would be bound to implementation of Coin

Method implementation using polymorphic parameter

```
public void add(Measurable x)
{
   sum = sum + x.getMeasure();
   if (count == 0 || maximum.getMeasure() < x.getMeasure())
      maximum = x;
   count++;
}
```

Client-code sending substitute

```
coinData.add(new Coin(0.25, "quarter"));
```

other parameter of BankAccount points to SavingAccount object studentFund

Method implementation using polymorphic parameter

```
public void transfer(double amount, BankAccount other)
{
    withdraw(amount);
    other.deposit(amount);
}
```

Client-code sending substitute

```
SavingsAccount studentFund = new SavingsAccount(10);
BankAccount collegeFund = new BankAccount(100000);
collegeFund.transfer(5000, studentFund);
```

Question: What should be output of this code?

```
public class A {
    public String str(){
        return "inside A";
    }
}
```

```
public class B extends A{
    public String str(){
        return "inside B";
    }
}
```

```
public class ABTester {
    private static void process(A a) {
        System.out.println(a.str());
    }
    public static void main(String[] args) {
        A a = new A();
        process(a);
        B b = new B();
        process(b);
    }
}
```

Use of polymorphic variables: to have collection of different type of objects

- You will find polymorphic variables are very often used as following. Here each array element is polymorphic variable of type Employee.

```
ArrayList<Employee> emps = new ArrayList<Employee>();
emps.add( new Engineer("Sumit Mehta", 45000.00, d1) );
emps.add( new Manager("Anil Kishore", 50000.00, d2) );
emps.add( new Staff("Suman Singh", 25000.00, d2) );

double total = 0;
for(int i = 0; i < emps.size(); i++) {
    System.out.println( emps[i] );
    total += emps[i].getSalary() + emps[i].getBonus();
}
System.out.println("Total Salary = " + total );
```

Polymorphism

- In Java, polymorphism is, changing behavior of a reference variables based upon its dynamic type (or object to which it points)

- That is basically polymorphic variable are behaving polymorphically.

- Polymorphism is accomplished in java by type substitution, and late binding

- Questions?