- A subclass inherits data and behavior from a superclass.
- You can always use a subclass object in place of a superclass object.
- A subclass inherits all methods that it does not override.
- A subclass can override a superclass method by providing a new implementation.
- The extends reserved word indicates that a class inherits from a superclass
- An overriding method can extend or replace the functionality of the superclass method.
- Use the reserved word super to call a superclass method.
- Unless specified otherwise, the subclass constructor calls the superclass constructor with no arguments.

## Syntax 9.1 Inheritance

```
Syntax class SubclassName extends SuperclassName {
    instance variables methods }

Example

Example

Subclass

Superclass

Superclass

Superclass

Superclass

Peclare instance variables f that are added to the subclass.

Peclare methods that are specific to the subclass.

Peclare methods that are specific to the subclass.

Superclass

Superclass

Superclass

Superclass

Superclass

Superclass

Public class SavingsAccount extends BankAccount

f private double interestRate; The reserved word extends demotes inheritance.

Peclare methods that are specific to the subclass.

Superclass

Sup
```

Big Java by Cay Horstmann Copyright © 2009 by John Wiley & Sons. All rights reserved.

- To call a superclass constructor, use the super reserved word in the first statement of the subclass constructor.
- The constructor of a subclass can pass arguments to a superclass constructor, using the reserved word super.
- Override the toString method to yield a string that describes the object's state.
- The equals method checks whether two objects have the same contents.
- If you know that an object belongs to a given class, use a cast to convert the type.
- The instance of operator tests whether an object belongs to a particular type.

## Example code: public class ChoiceQuestion

```
public void display()
{

// Display the question text

// Display the answer choices

for (int i = 0; i < choices.size(); i++) {

int choiceNumber = i + 1;

System.out.println(choiceNumber + ": " + choices.get(i)); }</pre>
```