Objectives

- In this session, you will learn to:
  - Understand the use of casting objects
  - Describe overloading methods and methods with variable arguments
  - Describe overloading constructors and invoking parent class constructors
  - Understand Wrapper classes
  - Understand autoboxing of primitive types
  - Create static variables, methods, and initializers

Casting Objects

- Casting objects is used where you have received a reference to a parent class, and you want to access the full functionality of the object of the subclass:
  - Use `instanceof` to test the type of an object.
  - Restore full functionality of an object by casting.
  - Check for proper casting using the following guidelines:
    - Casts upward in the hierarchy are done implicitly.
    - Downward casts must be to a subclass and checked by the compiler.
    - The object type is checked at runtime when runtime errors can occur.

Methods Using Variable Arguments

- The `vararg` or variable arguments is a feature provided by J2SE 5.0.
- It helps to pass variable number of arguments, of the same type, as parameters, to a method.
- It can be used when you have a number of overloaded methods, which share the same functionality.

Methods Using Variable Arguments (Contd.)

- The following example demonstrates the usage of varargs:

```
public class Statistics {
    public float average(int... nums) {
        int sum = 0;
        for ( int x : nums ) {
            sum += x;
        }
        return ((float) sum) / nums.length;
    }
}
```

nums is an array of type int[]

- We can invoke the average method by passing any number of arguments as integers.

Overloading Constructor

- As with methods, constructors can be overloaded:
  - An example is:

```
public Employee(String name, double
salary,Date doB)
public Employee(String name, double salary)
public Employee(String name, Date DoB)
```

  - Argument lists must differ.
  - You can use the this reference at the first line of a constructor to call another constructor.

Constructors Are Not Inherited

- A subclass inherits all methods and variables from the superclass (parent class).
- A subclass does not inherit the constructor from the superclass.
- Two ways to include a constructor are:
  - Use the default constructor
  - Write one or more explicit constructors

Invoking Parent Class Constructors

- To invoke a parent class constructor, you must place a call to `super()` in the first line of the constructor.
- You can call a specific parent constructor by the arguments that you use in the call to `super()`.
- If the parent class defines constructors, but does not provide a no-argument constructor, then a compiler error message is issued.

The Object Class

- The `Object` class is the root of all classes in Java.
- A class declaration with no extends clause implies extends `Object`. For example:

```
public class Employee
{
...
}
```

is equivalent to:

```
public class Employee extends Object
{
...
}
```

- Two important methods of object class are:
  - `equals()`
  - `toString()`

The equals Method

- The == operator determines if two references are identical to each other (that is, refer to the same object).
- The `equals()` method determines if objects are equal but not necessarily identical.
- The Object implementation of the `equals()` method uses the == operator.
- User classes can override the equals method to implement a domain-specific test for equality.

Note: You should override the `hashCode` method if you override the `equals` method.

The toString Method

- The `toString()` method has the following characteristics:
  - This method converts an object to a `String`.
  - Use this method during string concatenation.
  - Override this method to provide information about a user-defined object in readable format.
  - Use the wrapper class's `toString()` static method to convert primitive types to a String.

Wrapper Classes

- The Java programming language provides `wrapper` classes to manipulate primitive data elements as objects.
- Each Java primitive data type has a corresponding wrapper class in the `java.lang` package.
- Example of Primitive Boxing using wrapper classes:

```
int pInt = 420;
Integer wInt = new Integer(pInt);
// this is called boxing
int p2 = wInt.intValue();
// this is called unboxing
```

Autoboxing of Primitive Types

- The autoboxing feature enables you to assign and retrieve primitive types without the need of the wrapper classes.
- Example of Primitive Autoboxing:

```
int pInt = 420;
Integer wInt  = pInt; // this is called autoboxing
int p2 = wInt;        // this is called autounboxing
```

- The J2SE 5.0 compiler will create the wrapper object automatically when assigning a primitive to a variable of the wrapper class type.
- The compiler will also extract the primitive value when assigning from a wrapper object to a primitive variable.

The static Keyword

- The `static` keyword is used as a modifier on variables, methods, and nested classes.
- The `static` keyword declares the attribute or method is associated with the class as a whole rather than any particular instance of that class.
- Thus,static members are often called class members, such as `class` attributes or `class` methods.

The static Keyword (Contd.)

- Static Attribute:
  - A public static class attribute can be accessed from outside the class without an instance of the class.
- Static Method:
  - A static method can be invoked without creating the instance of the class.
  - Static methods can not access instance variables.
- Static Initializers:
  - A class can contain code in a static block that does not exist within a method body.
  - Static block code executes once only, when the class is loaded.
  - Usually, a static block is used to initialize static (class) attributes.

Summary

- In this session, you learned that:
  - Casting objects is used where you have received a reference to a parent class, and you want to access the full functionality of the object of the subclass.
  - The methods which perform closely related tasks can be given the same name by overloading them.
  - The `varargs` feature helps us to write a generic code to pass variable number of arguments, of the same type to a method.
  - The `super` keyword is used to call the constructor of the parent class.
  - The `object` class is the root of all classes. The two important methods of the object class are `equals()` method and `tostring()` method.

Summary (Contd.)

- Wrapper classes are used to manipulate primitive data elements as objects. Each Java primitive data type has a corresponding `wrapper` class in the `java.lang` package.
- Autoboxing feature of J2SE 5.0 enables you to assign and retrieve primitive types without the need of the wrapper classes.
- The `static` keyword declares members (attributes, methods, and nested classes) that are associated with the class rather than the instances of the class.