COJ :: Getting Started With Java

TalentSprint

Licensed To Skill

Version 1.0.4

Classes, Objects

The content in this presentation is aimed at teaching learners to:

- Define class and object
- Differentiate class and object
- Create simple java classes, construct and use java objects

Classes, Objects

The Dice Game

Player 1

Dice 1

Dice 2

Player 2

Classes, Objects

An object is some **real** or conceptual thing.

What is interesting about dice?

Its face value

What does a dice do?

It rolls.

What happens when a dice rolls?

Its face value changes.

An object is a self-contained entity that consists of both data (properties) and methods (behaviour) to manipulate the data.

Classes, Objects

Properties & behaviors of Dice Objects

Dice 1 has a face value
Dice 2 also has a face value
.
.
.
Dice n also has a face value

Dice 1 rolls
Dice 2 also rolls
.
.
.
Dice n also rolls

- Any dice has: A face value
- Any dice: Rolls

In other words,

- Any object that is a dice has a face value and rolls.

In other words,

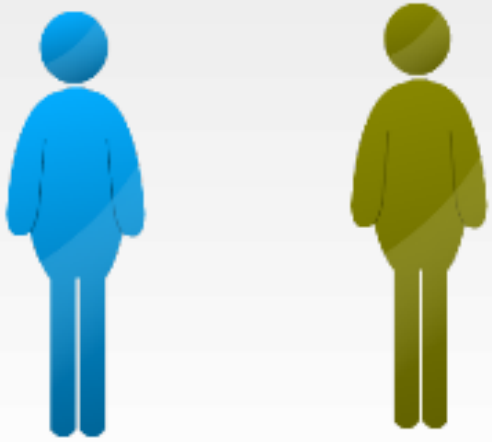
- An object if it is of type dice has a face value and rolls.

*An object if it is of **class** dice has a face value and rolls.*

Classes, Objects

Classes in Dice Game

Player Class



Dice Class

Classes, Objects

A class is a description of a set of objects that share the same attributes and behaviour (operations)

Classes, Objects

Recall the following definitions of abstraction and explain the connect between abstraction and class:

What is class?

Classes, Objects

List out the properties & behaviors of Player class.
Hint : What is interesting about player?
What does a player do?

Classes, Objects

Properties & Behaviors of Classes in Dice Game

Dice

Properties
faceValue
Behaviors
roll

Player

Properties
name
value
Behaviors
throw

Classes, Objects

Properties & Behaviors of Dice Class

Dice

Properties

faceValue

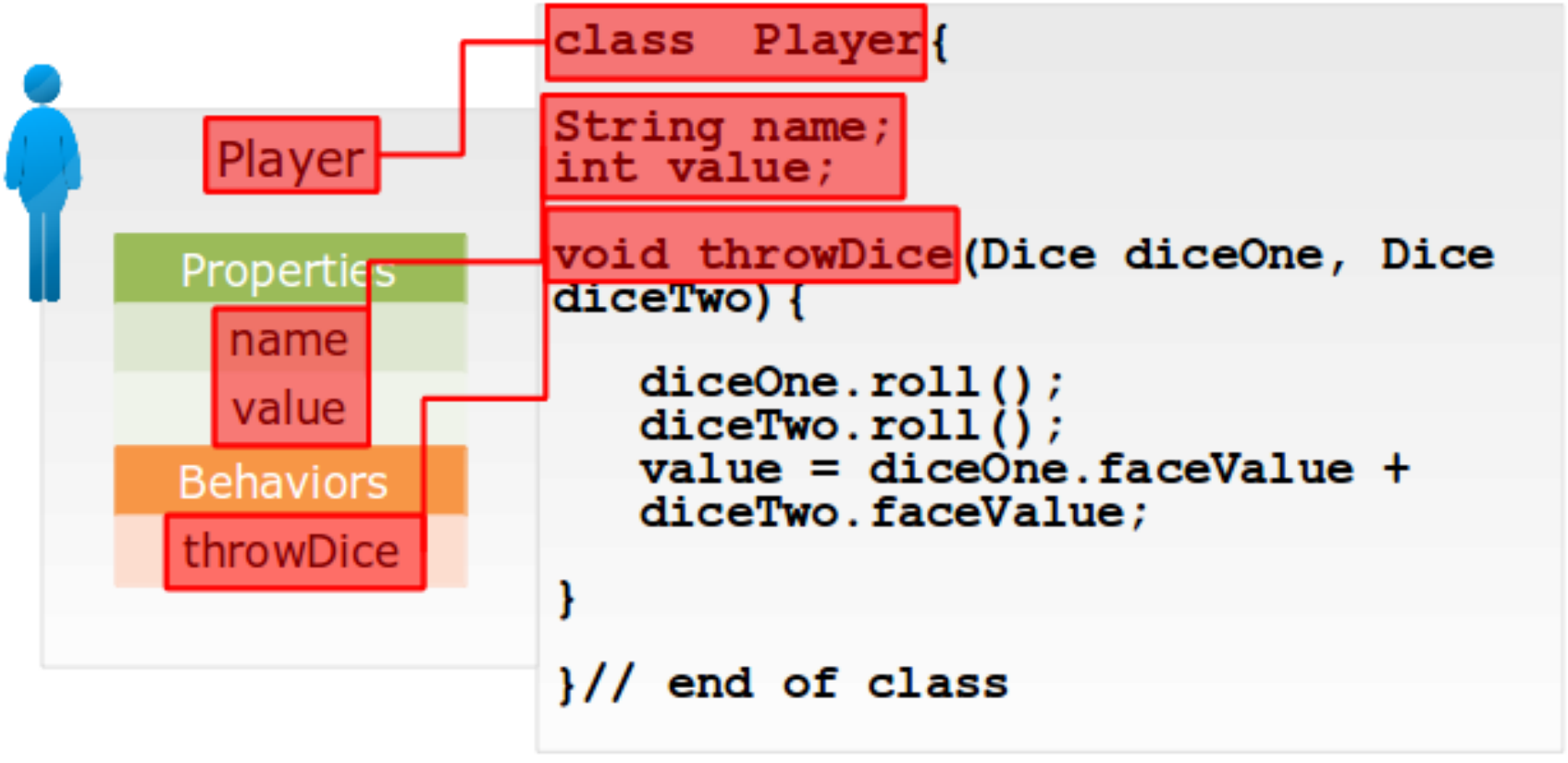
Behaviors

roll

```
class Dice{

int faceValue;

void roll(){
//random no. between 1 and 6
   faceValue                    =(int)
   ((Math.random())*10)%5+1;
}//  end of roll

}// end of class
```

Classes, Objects

Properties & Behaviors of Player Class



Classes, Objects

Programmatic definition of Class

A class is a representation for a set of objects that are data abstractions with an interface of named operations (methods) and hidden local state (attributes).

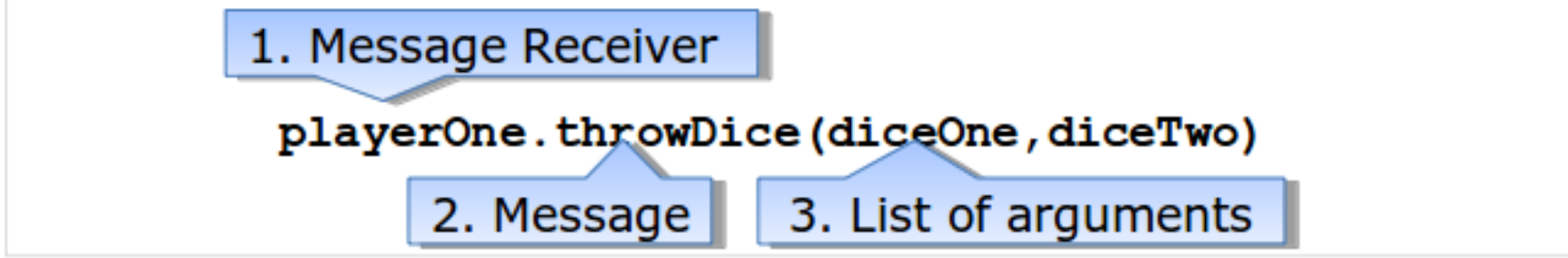
Classes, Objects

Behavior in an O-O program

- In Object-oriented programming, programs are organized as cooperative collections of objects.
- Each object represents an instance of some class.
- Objects communicate by passing messages (by calling methods).
  - A message is always given to some object.
  - The response to a message depends upon the class of the Object.

Classes, Objects

- All messages have three identifiable parts.



Classes, Objects

Making Objects Collaborate

Steps in making objects collaborate with each other:

Step 1

Define Classes

Step 2

Create Objects

Step 3

Pass Messages

Classes, Objects

Dice Game Class and main method

```
class DiceGame {
    public static void main (String args [] ) {
        Dice diceOne;
        diceOne = new Dice();
        // create diceTwo object
        Player playerOne;
        playerOne = new Player();
        playerOne.throwDice(diceOne,diceTwo);
```

Classes, Objects

```
        // create playerTwo object
        // ask playerTwo to throwDice
        if  (playerOne.value > playerTwo.value) {
             System.out.println("Player1 Wins");
        }
        else {
             System.out.println("Player2 Wins");
        }
    }// end of main(
}//end of class DiceGame
```

Classes, Objects

Defining a Class

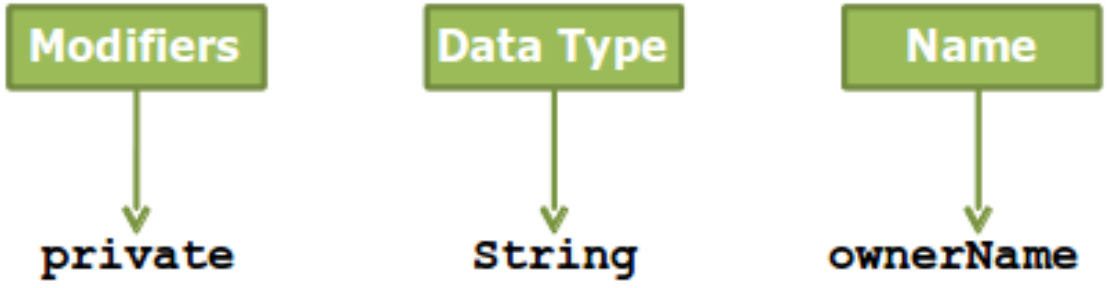
```
class classname {
type instance-variable1;
type instance-variable2;
// ...
type instance-variableN;
type methodname1(parameter-
list) {
// body of method
}
type methodname2(parameter-
list) {
// body of method
}
// ...
type methodnameN(parameter-
list) {
// body of method
}
}
```

Example :

```
class Dice{

int faceValue;

void roll(){

     faceValue  =  (int)
((Math.random())*10)%5 + 1;

}//  end of roll

}// end of class
```

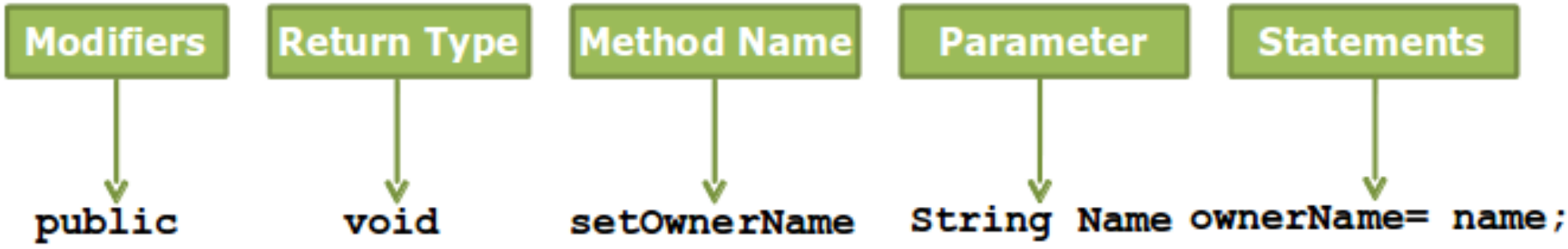
Classes, Objects

<modifier> <data type> <name>



Classes, Objects

<modifier> <return type> <method name> (<paramaters>) { <statements> }



Classes, Objects

Creating Objects

```
ClassName refVariable;
refVariable = new
              Constructor();
      or
ClassName  refVariable =
          new  Constructor();
```

Example :

```
Example :
Dice diceOne ;
diceOne = new Dice();
                  or
Dice diceTwo = new Dice();
```

Classes, Objects

Using Objects

```
Dice diceOne = new Dice();
diceOne.roll();
int value = diceOne.faceValue();
```

Classes, Objects

Using Objects

There is one more object in the game. It is the diceGame object itself. Because every object in Java has to belong to some class, let's define the DiceGame class.

An object is some real or conceptual thing.

Classes, Objects

Properties & Behaviors of Dice Game Classes


```
class DiceGame {
    Player playerOne, playerTwo;
    Dice Diceone, DiceTwo;
    void play() {
        diceOne = new Dice();
        diceTwo = new Dice();
```

Classes, Objects

```
        playerOne = new Player();
        playerTwo = new Player();
        playerOne.throwDice(diceOne,diceTwo);
        playerTwo.throwDice(diceOne,diceTwo);
        if  (playerOne.value > playerTwo.value) {
            System.out.println("Player  One Wins");
        }
        else {
            System.out.println("Player  Two Wins");
        }
    }// end of play()

}//end of class
```

Classes, Objects

Create the main method for the DiceGame class and complete the game.

Classes, Objects

Using Objects

```
Dice diceOne = new Dice();
diceOne.roll();
int value =
diceOne.getFaceValue();
```

Classes, Objects