Classes and Objects

Class is a template for an object(contains the data fields and methods)

```
public class Employee{
double salary;
double bonus;

Void calculateTotalPay(){

double totalpay=salary+bonus;
System.out.println("Totalpay is"
+totalpay);

}
}
```

Data fields

method

Object is a copy of a class(instance of a class)

```
public class TestEmployee{
public static void main (String[]
args){
Employee alex =new Employee();
Employee linda =new Employee();
alex.salary=90000;
alex.bonus=20000;
alex.calculateTotalPay();
}
}
```

Classes and Objects

```
public class Box{
int length;
int width;

Void calculateArea(){

int area=length*width;
System.out.println("Area is" +area);

}
}
```

```
public class TestBox{
public static void main (String[]
args){
Box obj1 =new Box();
Box obj2 =new Box();
obj1.length=5;
obj1.width=10;
obj1.calculateArea;
obj2.length=15;
obj2.width=2;
obj2.calculateArea;
}
}
```

Advantage of return method calculating sum of areas

```
public class Box{
int length;
int width;

Int  calculateArea(){

int area=length*width;
return area;

}
}
```

```
public class TestBox{
public static void main (String[]
args){
Box obj1 =new Box();
Box obj2 =new Box();
obj1.length=5;
obj1.width=10;
Int area1=obj1.calculateArea;
obj2.length=15;
obj2.width=2;
Int area2=obj2.calculateArea;
System.out,println(area1+area2);
}
}
```

Methods with arguments

```
public class Box{
int length;
int width;

Int  calculateArea(int x){

int area=length*width*x;
return area;

}
}
```

```
public class TestBox{
public static void main (String[]
args){
Box obj1 =new Box();
Box obj2 =new Box();
obj1.length=5;
obj1.width=10;
Int area1=obj1.calculateArea(4);
System.out.println(area1);
}
}
```

Methods with arguments

```
public class Box{
int length;
int width;

Int  calculateArea(int length,int
width){

int area=length*width;
return area;

}
}
```

```
public class TestBox{
public static void main (String[]
args){
Box obj1 =new Box();
Box obj2 =new Box();
obj1.length=5;
obj1.width=10;
Int area1=obj1.calculateArea(4,3);
System.out.println(area1);
}
}
```

Output 12

Methods with arguments

```
public class Box{
int length;
int width;

Int  calculateArea( int length,int
width){

int area=this.length*this.width;
return area;

}
}
```

```
public class TestBox{
public static void main (String[]
args){
Box obj1 =new Box();
Box obj2 =new Box();
obj1.length=5;
obj1.width=10;
Int area1=obj1.calculateArea(4,3);
System.out.println(area1);
}
}
```

Output 50

Constructors

```
public class SmallBox{
int length;
int width;

SmallBox(int  length, int width) {
this.length = length;
this.width=width;
}

void  calculateArea( ){

System.out.println("Area is" +(length+width));

}
}
```

```
public class TestSmallBox{
public static void main (String[]
args){
SmallBox obj1 =new SmallBox(3,4);
obj1.calculateArea();
}
}
```

Output ?

Constructors

```
public class SmallBox{
int length;
int width;

SmallBox( ){

this.length=5;
this.width=6;
}

void calculateArea(){
System.out.println("Area"+(length+width)
);
}
}
```

```
public class TestBox{
public static void main (String[]
args){
SmallBox obj1 =new Employee();
obj1.calculateArea();
}
}
```

Output ?

if you want some defalt value you can use a constructor(initial defalt values)

Concructors have the same name as class

Doesn't't return anything including void

Constructors with parameters

```
public class SmallBox{
int length;
int width;

SmallBox(int length,int width ){

this.length=length;
this.width=width;
}

void calculateArea(){
System.out.println("Area"+(length+width)
);
}
}
```

```
public class TestBox{
public static void main (String[]
args){
Box obj1 =new Employee();
obj1.calculateArea();
}
}
```

Output ?

if you want some defalt value you can use a constructor(initial defalt values)

Concructors have the same name as class

Doesn't't return anything including void

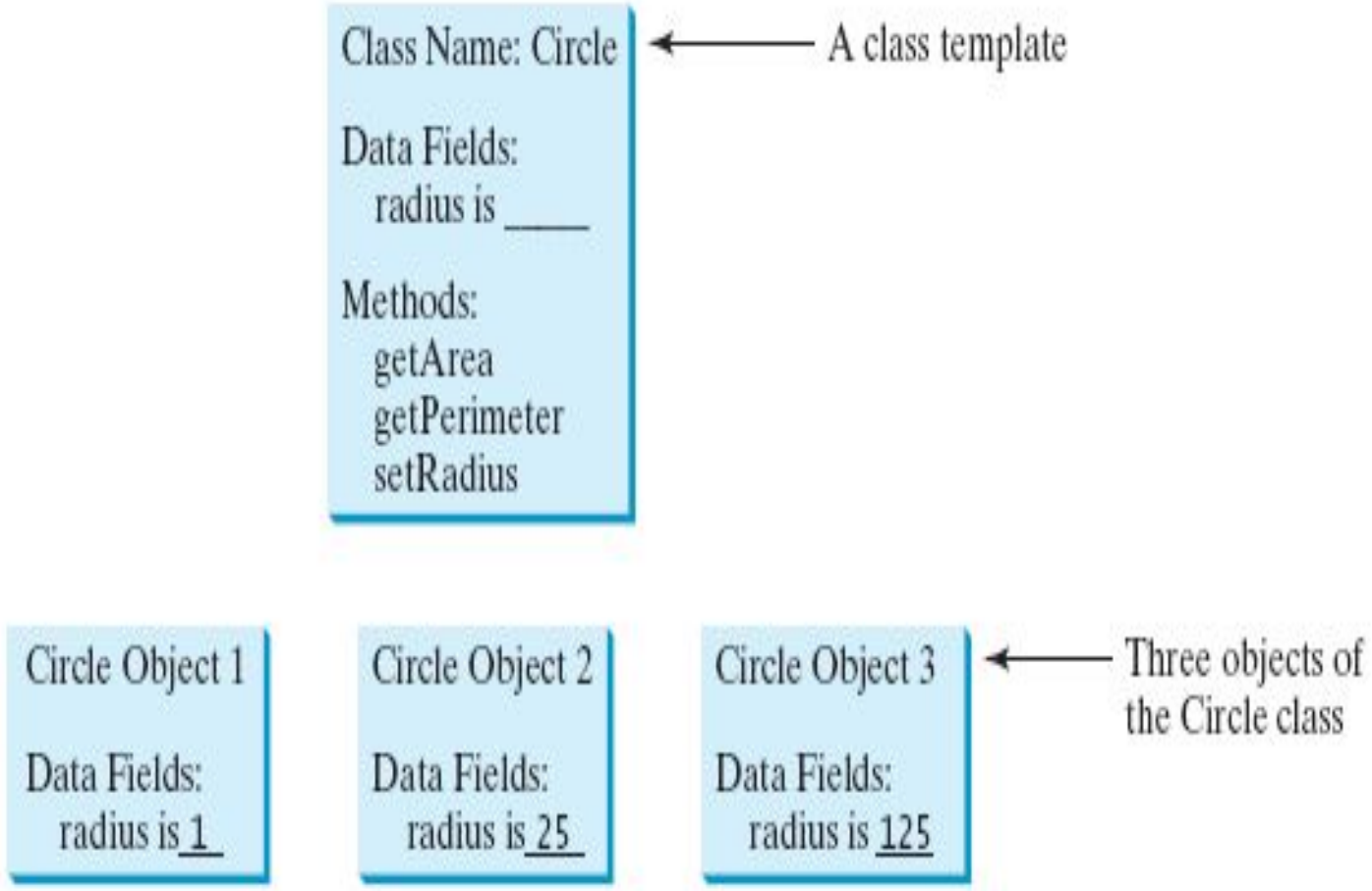

FIGURE 8.2 A class is a template for creating objects.