

# Pitfalls

## 1. Returning the wrong type.

```
static int area(int base, int height) {  
    return "area is: " + base + " * " + height;  
}
```

...

Triangle.java:9: incompatible types

found : java.lang.String

required: int

```
    return "area is: " + base + " * " + height;  
                                ^
```

1 error

# Pitfalls

## 2. Going against explicit cast in the method.

```
static int area(int base, int height) {  
    return ((double) base * height) / 2;  
}
```

...

```
Triangle.java:9: possible loss of precision  
found   : double  
required: int  
    return ((double) base * height) / 2;  
                ^
```

1 error

# Pitfalls

## 3. Going against explicit casts on the caller side.

```
System.out.println(area(4.0));  
...  
static double area(int equilateralSide) {  
    return equilateralSide * equilateralSide * .433;  
}
```

...

```
Triangle.java:5: area(int) in PitFall cannot be applied to (double)  
location: class Triangle
```

```
System.out.println(area(4.0));  
                        ^
```

1 error

# Pitfalls

## 4. Returning a value from a void method.

```
static void printArea(int base, int height) {  
    return (base * height) / 2;  
}
```

...

Triangle.java:18: cannot return a value from method whose result type is void

```
    return (base * height) / 2;  
    ^
```

1 error

# Pitfalls

## 5. Trying to use a value from a void method.

```
        System.out.println(area(5, 10));  
        ...  
static void area(int base, int height) {  
    System.out.println((base * height) / 2);  
}
```

...

Triangle.java:3: 'void' type not allowed here

```
    System.out.println(area(5, 10));  
                        ^
```

1 error

# Pitfalls

## 6. Double declaration.

```
static int area(int base, int height) {  
    return (base * height) / 2;  
}  
static int area(int side1, int side2) {  
    return side1 * side2;  
}
```

...

Triangle.java:11: area(int,int) is already defined in Triangle

```
static int area(int side1, int side2) {  
    ^
```

1 error

## Pitfalls

### 7. Ambiguous use of overloading.

```
        System.out.println(area(5, 10));  
    ...  
    static double area(int base, double height) {  
        return (base * height) / 2;  
    }  
    static double area(double base, int height) {  
        return (base * height) / 2;  
    }
```

...

Triangle.java:3: reference to area is ambiguous, both method  
area(int,double) in Triangle and method area(double,int) in  
Triangle match

```
        System.out.println(area(5, 10));  
                                ^
```

1 error

# Pitfalls

## 8. Overloading on the return type.

```
static int area(int base, int height) {  
    return (base * height) / 2;  
}  
static double area(int base, int height) {  
    return ((double) base * height) / 2;  
}
```

...

Triangle.java:11: area(int,int) is already defined in Triangle

```
    static double area(int base, int height) {  
        ^
```

1 error