Casting

COMP 102

Victoria University of Wellington

Types and Coercion

- Mismatching types:

```
double num = scan.nextInt( );          ← No information lost
int number = scan.nextDouble( );       ← Can't do this without loosing information, Error
double squareroot  =  Math.sqrt(25);   ← but sqrt wants double?
String name = "number-" + num;         ← Objects and primitives can be translated to text
```

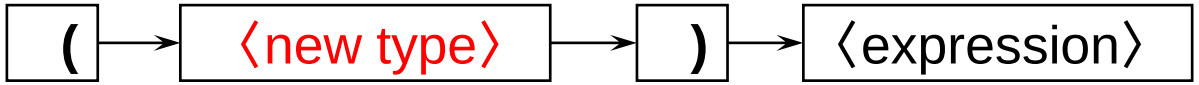
- Java will "coerce" a value to the needed type if it can: eg
  - If a method needs a double and is given an int.
  - If a double variable is assigned an int value.
  - If "adding" any value to a String
  - converting between double and Double or int and Integer (or the other Wrapper Types)

- But only if it does not lose any information:
  - WON'T coerce a double to an int
  - WON'T coerce a String to a number, or vice versa (except when "adding" a number to a String)
  - WON'T coerce any object to a mismatching type (except when printing or "adding" to a String)

Casting

- Where it makes sense to convert a value into another type, but some information may be lost...
- You can *sometimes* "cast" the value to the other type:

```
int number = (int) Math.sqrt(49.5);
float red =  (float) Math.random();
```



- casting a double to an int will lose the fractional part and may mess up the value if the number is too big!
- Not everything can be cast to everything else!
  - Scanner scan = (Scanner) (new PrintStream("data.txt"));
  - ButterFly b1 = (ButterFly) (new Lollipop(100,100,40,50));