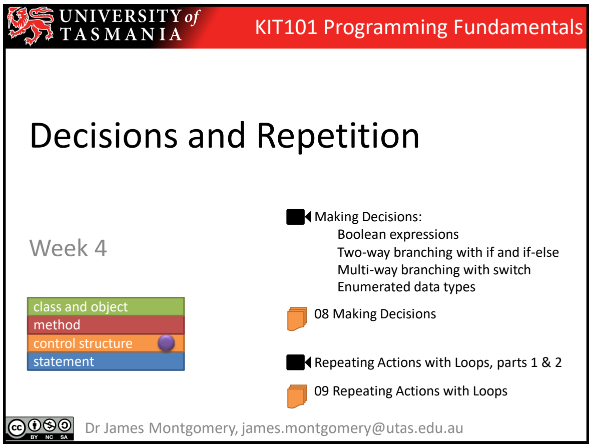


UNIVERSITY of TASMANIA — KIT101 Programming Fundamentals

Decisions and Repetition

Week 4

- class and object
- method
- control structure
- statement

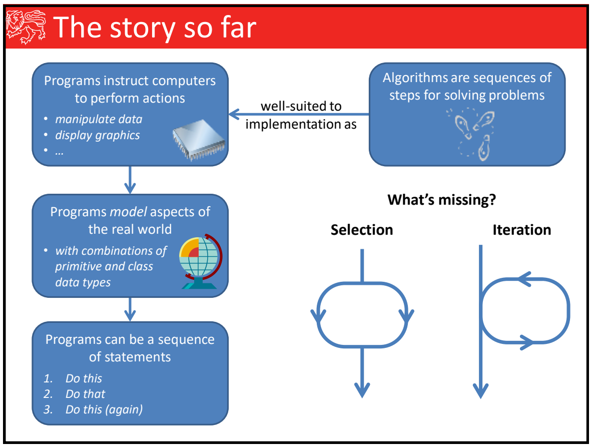
Making Decisions:
Boolean expressions
Two-way branching with if and if-else
Multi-way branching with switch
Enumerated data types

08 Making Decisions

Repeating Actions with Loops, parts 1 & 2

09 Repeating Actions with Loops

Dr James Montgomery, james.montgomery@utas.edu.au



The story so far

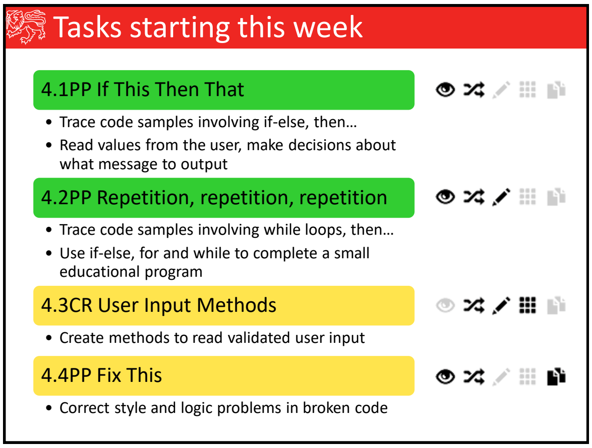
Programs instruct computers to perform actions
- *manipulate data*
- *display graphics*
- *…*

Programs *model* aspects of the real world
- *with combinations of primitive and class data types*

Programs can be a sequence of statements
1. *Do this*
2. *Do that*
3. *Do this (again)*

Algorithms are sequences of steps for solving problems

What's missing?



Tasks starting this week

4.1PP If This Then That
- Trace code samples involving if-else, then…
- Read values from the user, make decisions about what message to output

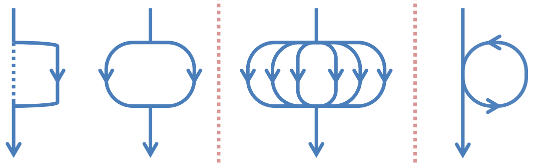
4.2PP Repetition, repetition, repetition
- Trace code samples involving while loops, then…
- Use if-else, for and while to complete a small educational program

4.3CR User Input Methods
- Create methods to read validated user input

4.4PP Fix This
- Correct style and logic problems in broken code

Control structures

if | if-else | switch | Loops while do-while for



Select a path based on value of a **Boolean expression**

Paths 'labelled' by a **value** (integer-valued primitives, Enums and Strings)

Repeat actions while **Boolean expression** is true

The simplest Boolean expression is ==

The == operator can be used to check for exact equality of

- integers
- characters
- floating-point numbers (be careful as they are not stored precisely)
- object *references*

==

Comparing… objects

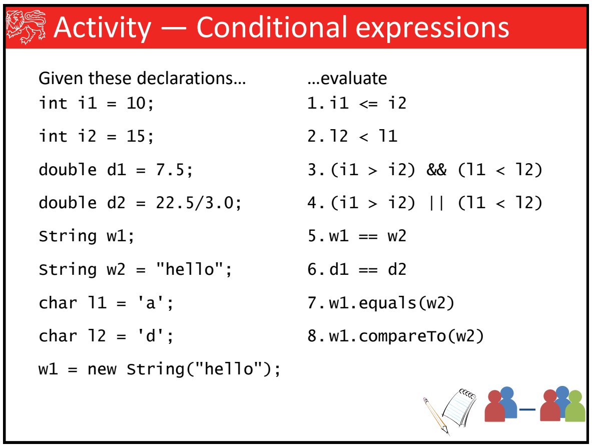
- All objects have this method:
  `public boolean equals(Object o)`
  - In some, no better than ==
  - In others, returns `true` if internal *state* is same
  - Usage: `someObject.equals( anotherObject );`

- Strings (and other Comparables) also have
  `public int compareTo(String s)`

```
"alpha".compareTo("zeta") == -25
"alpha".compareTo("alpha") == 0
"zeta".compareTo("alpha") == +25
```

Any result < 0 means the first object belongs before the second, any result > 0 means it belongs after it

see StringComparisons.java



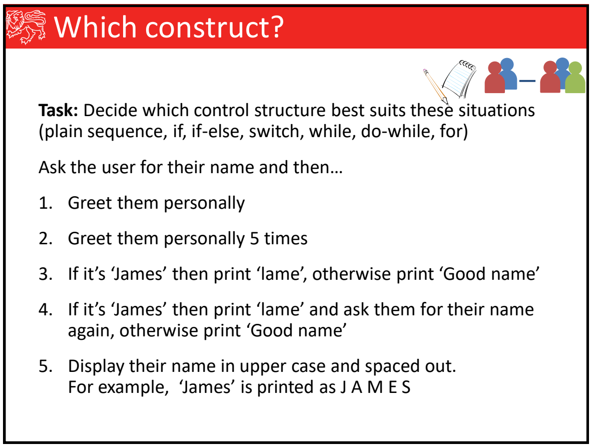
Activity — Conditional expressions

Given these declarations…

```
int i1 = 10;
int i2 = 15;
double d1 = 7.5;
double d2 = 22.5/3.0;
String w1;
String w2 = "hello";
char l1 = 'a';
char l2 = 'd';
w1 = new String("hello");
```

…evaluate

```
1. i1 <= i2
2. l2 < l1
3. (i1 > i2) && (l1 < l2)
4. (i1 > i2) || (l1 < l2)
5. w1 == w2
6. d1 == d2
7. w1.equals(w2)
8. w1.compareTo(w2)
```

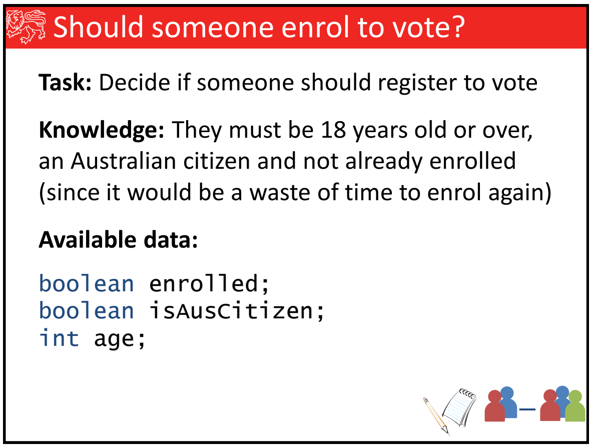


Which construct?

Task: Decide which control structure best suits these situations (plain sequence, if, if-else, switch, while, do-while, for)

Ask the user for their name and then…

1. Greet them personally
2. Greet them personally 5 times
3. If it's 'James' then print 'lame', otherwise print 'Good name'
4. If it's 'James' then print 'lame' and ask them for their name again, otherwise print 'Good name'
5. Display their name in upper case and spaced out.
   For example, 'James' is printed as J A M E S



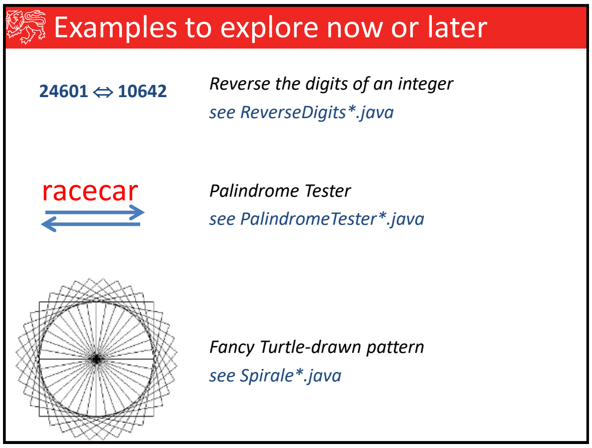
Should someone enrol to vote?

Task: Decide if someone should register to vote

Knowledge: They must be 18 years old or over, an Australian citizen and not already enrolled (since it would be a waste of time to enrol again)

Available data:

```
boolean enrolled;
boolean isAusCitizen;
int age;
```



Examples to explore now or later

24601 ⇔ 10642

Reverse the digits of an integer
see ReverseDigits.java*

racecar

Palindrome Tester
see PalindromeTester.java*

Fancy Turtle-drawn pattern
see Spirale.java*