Testing

COMP 102

Victoria University of Wellington

Testing your program

A) Need to try out your program on sample input while removing the "easy" bugs.

- Can be a pain if need lots of input
- UI window has a menu item – "set input" – to get input from a text file instead of user typing it.

  ⇒ don't have to type lots of data each time

  - Create the text file, eg in Notepad
  - Select file using menu *before* the program has started asking for input.
  - File can contain multiple sequences of data.

Testing your program

A) Need to try out your program on sample input while removing the "easy" bugs.

- Can be a pain if need lots of input
- UI window has a menu item – "set input" – to get input from a text file instead of user typing it.

  ⇒ don't have to type lots of data each time
  - Create the text file, eg in Notepad
  - Select file using menu *before* the program has started asking for input.
  - File can contain multiple sequences of data.

- B) Need to test your program on a range of inputs
  - Easy, "ordinary", inputs
  - Boundary cases — values that are only just in range, or just out of range
    Need to check that your **if** conditions are right
  - Invalid data—does your program handle invalid input correctly?

  Creating test cases involves creativity – have to try to come up with ways to break your program.

What are boundaries for valid emails?

```
/** Ask for an email address and insist that it contains one @ in the middle */
public String askEmailAddress (){
    String addr = UI.askString("Enter email address");
    while ( ! this.validEmailAddress(addr) ){
        UI.println("Your email is invalid");
        addr = UI.askString("Enter email address");
    }
    return addr;
}

/** Check that an email address contains one @ in the middle and no spaces */
public boolean validEmailAddress (String addr){
    int idx = addr.indexOf("@");
    return (idx>0 && idx!=addr.length()-1&& addr.indexOf("@", idx+1)==-1 && !addr.contains(" "));
}
```