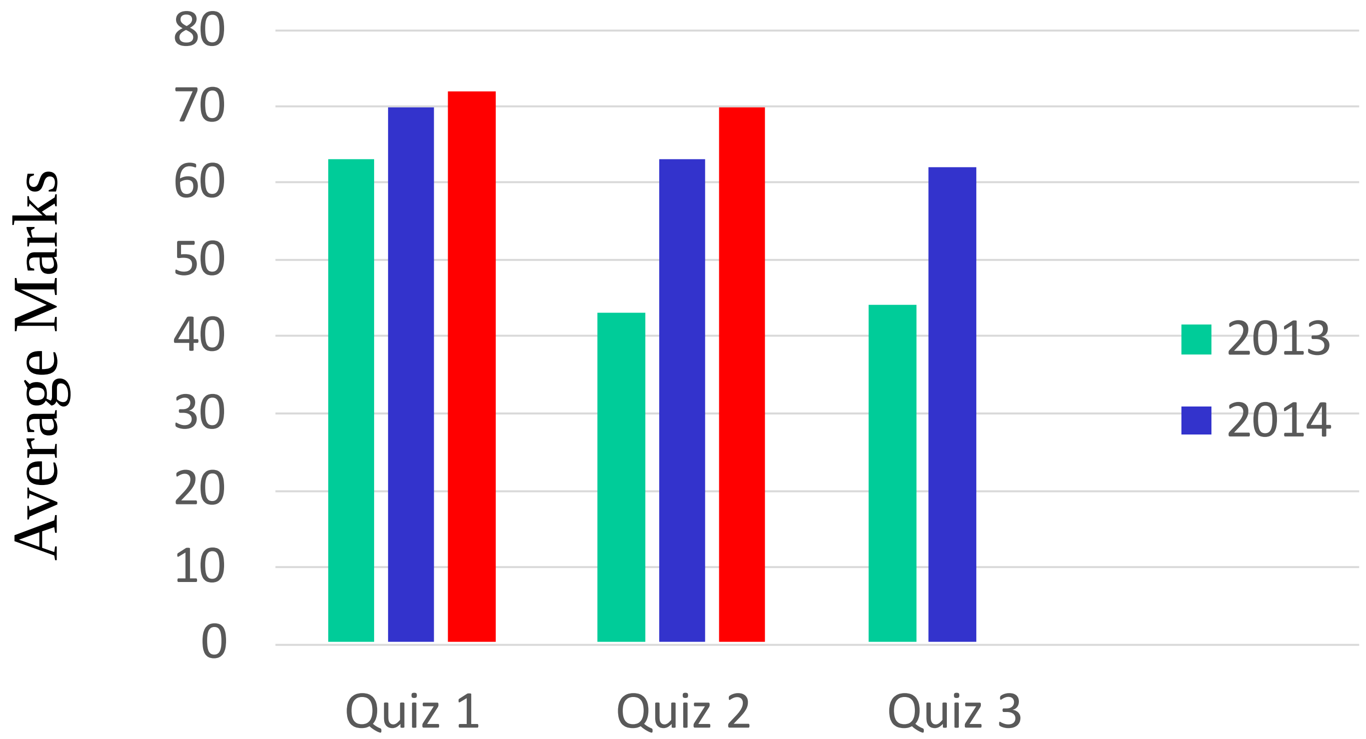
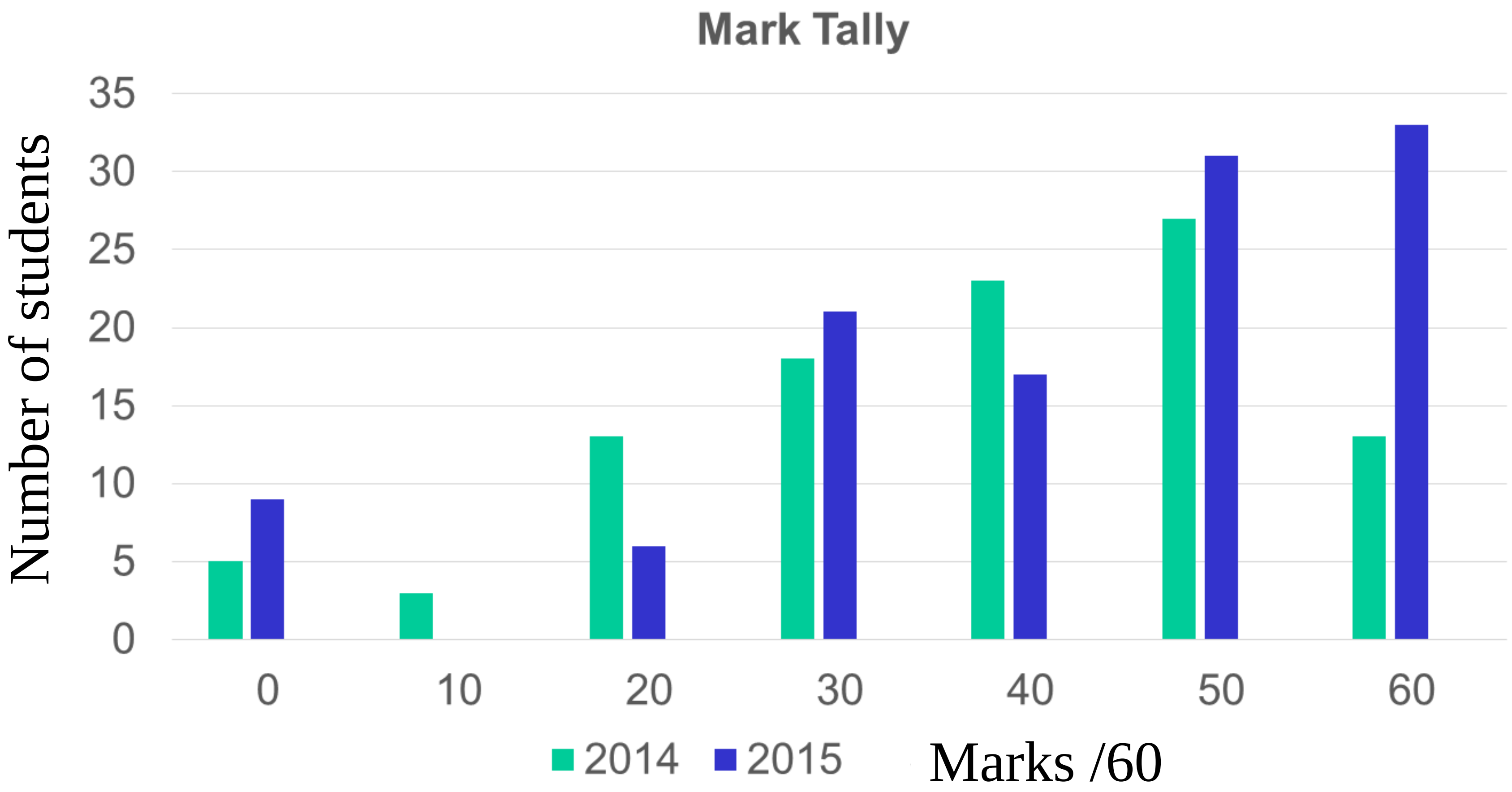
Mole Quiz 2: Results

Quiz Results 2013, 2014, 2015



Improvement on last year

Significant improvement compared to year before



- Signifcant increase in numbers achieving full marks. (**13/102 up to 32/119**)
- Significant decrease in number failing (**16/102 down to 6/109**)
- Increase in number not taking test (**5/102 up to 9/119**)

Email me if you want extra tutorial before Quiz 3

4 people: Quiz 1 0; Quiz 2 0

5 people: Quiz 1 (pass); Quiz 2 0.

6 people: Quiz 2 20/60

(Quiz 1 (/90): 90; 59, 82.5, 69, 80, 58)

21 people: 30/60

17 people: 40/60

31 people: 50/60

33 people: 60/60

Difficult to satisfy everyone:

- Too easy?
- Should negative marking be re-introduced ?
- More questions ? Less time ?
- Written exam ? (Increase percentage on exercise sheets)
- Faster paced lectures covering more material and more challenging material?

If we have a propositional function P(n), and we want to prove that P(n) is true for any natural number n, we do the following:

- Show that **P(1) is true.**
  (basis step)
- Show that if P(k) then P(k + 1) for any k∈N.
  (inductive step)
- Then P(n) must be true for any n∈N.
  (conclusion)

$$\left[P(1) \wedge \left(P(k) \Rightarrow P(k+1)\right)\right] \Rightarrow \forall n P(n)$$

Proof By Induction - Continued

1. Prove for all n>=1

$$\sum_{r=1}^{n} r(r+1) = \frac{1}{3}n(n+1)(n+2)$$

2. Prove for all n>=3

$$n^2 \geq 2n+1$$

Proofs done in lectures for those that attended.

Induction and Recursion

Learning Outcome

Define all the words (ie strings) of up to 3 character in length constructed from the letters a, b, c. These words need to have a particular form, which is defined by the following BNF grammar, where S denotes the permitted word structures.

S ::= a | b | c | aS | Sb | cS |

Motivation

In modelling software:

- ❑ Many structures are defined inductively or recursively:
  - ❖both for data and for program code.

In studying the behaviour of software:

- ❑ we need to use inductive and recursive structures;
- ❑ we need to reason about their properties.

Inductive Functions

Defining Functions Inductively:

For functions over inductive data types:

- ❖define for the basis clause(s), and
- ❖define for the inductive clause(s),
  - – in terms of the definition for the components;

❑Examples:

- ❖ factorial: $0! = 1$, and $n! = n \times (n-1)!$
- ❖ add: add (m, 0) = m,

  and

  add (m, n+1) = add (m, n) + 1.

Fibonacci numbers

The **Fibonacci numbers** are defined inductively as follows

$$f_0 = 0$$
$$f_1 = 1$$
$$f_n = f_{n-1} + f_{n-2} \quad (\text{for } n > 2)$$

Basic Concepts 1

Defining Sets:

- **Finite sets** can be defined easily:
  - by listing their elements;
- This is not possible for infinite sets:
  - we use the inductive approach instead;
- An inductive definition has three parts:
  - **a basis clause**, for the “starting” elements,
  - **an inductive clause**, to build other elements,
  - **an extremal clause**, to eliminate unwanted elements.

Basic Concepts 2

Simple example:

- The natural numbers $\mathbf{N}$:
  - basis clause: $0 \in \mathbf{N}$,
  - inductive clause: if $n \in \mathbf{N}$ then $n+1 \in \mathbf{N}$,
  - extremal clause: $\mathbf{N}$ has no other elements,
  - these three rules are called the **Peano axioms**;
- Alternatively, $\mathbf{N}$ is the smallest set where:
  - $0 \in \mathbf{N}$ and if $n \in \mathbf{N}$ then $n+1 \in \mathbf{N}$.

Basic Concepts 3

Syntactic Sets:

- we need to distinguish:
  - sets of names of objects, from
  - sets of the objects themselves:
    - ie *syntactic objects* from *semantic objects* (values/interpretations);
- **Syntactic sets** can be defined in terms of:
  - an **alphabet** of symbols (characters),
  - **words** (strings) - sequences of symbols,
    - including the empty word, denoted ε,
  - and **sentences** - sequences of words.

Basic Concepts 4

Sets of Words:

- A^* is the set of **words** over the **alphabet** A:
  - so A^* is the smallest set such that
  - $\varepsilon \in A^*$, and if $w \in A^* \wedge a \in A$ then $aw \in A^*$,
    - – this use of * is sometimes called the **Kleene star**;
- **A^+** is the set of **non-empty** words over A;
  - so A^+ is the smallest set such that
  - $a \in A \Rightarrow a \in A^+$, and
  - if $w \in A^+ \wedge a \in A$ then $aw \in A^+$.

Basic Concepts 5

Backus-Naur Form (BNF):

- a notation for writing inductive definitions:
  - as equations of the form:
    - **set_name :: = basis clauses | inductive clauses**
- e.g for A^*:
  - w ::= ε | aw
- e.g for A^+:
  - w ::= a | aw
- e.g for **N**:
  - n ::= 0 | successor (n)

Define the set S2 which consists of all the words (ie strings) of up to 2 character in length constructed from the letters a, b, c. These words need to have a particular form, which is defined by the following BNF grammar, where S denotes the permitted word structures.

S ::= a | b | c | aS | Sb | cS |

Solution

S2={aa,ab,ca,bb,ab,cb,ac,cb,cc}

={aa,ab,ca,bb,cb,ac,cc}

Define all the words (ie strings) of up to 3 character in length constructed from the letters a, b, c. These words need to have a particular form, which is defined by the following BNF grammar, where S denotes the permitted word structures.

$$S ::= a \mid b \mid c \mid aS \mid Sb \mid cS \mid$$

BNF grammars

The set of propositional formulae can be defined inductively as the smallest set satisfying the following:

1. True and false are propositional formulae, as is every propositional variable P
2. If p and q are propositional formulae then so are

$$\neg p, p \vee q, p \wedge q, p \Rightarrow q \quad \text{and} \quad p \Leftrightarrow q$$

$$\phi ::= true \mid false \mid P \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \Rightarrow \phi \mid \phi \Leftrightarrow \phi$$

BNF grammars: Predicate Logic

Give an inductive definition of the set of formulae of predicate logic.

$$\phi ::= true \mid false \mid P(x_1,\ldots,x_n) \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \Rightarrow \phi \mid \phi \Leftrightarrow \phi \mid \forall x\phi \mid \exists x\phi$$