Year 13 HL

Homework 1: Recursion

1) Describe what ‘recursion’ is.

2) Look at the following pseudocode:

```
function (n)
{
        if (n == 1)
                then return 2
                else return function (n-1) + $2^n$
}
```

a) Work out manually f(3).
b) Write the base case and the general case.
c) For the above recursive function to work a condition must be obeyed. What is it?
d) Write function (n) in Java and test it.
e) Also write an iterative method to solve the same problem.

3) Look at the following pseudocode:

```
stringFun (s)
{
        if (s has only one character)
                then return s
                else
                {
                        ll = last character of s
                        rest = s with ll removed
                        return ll + stringFun (rest)
                }
}
```

a) Work out manually stringFun (“abcd”).
b) What does stringFun (s) do?
c) Write stringFun (s) in Java and test it.
d) Also write an iterative solution for stringFun (s)

4) The Fibonacci numbers are the numbers in the following integer sequence.
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, … The first two terms are equal to 0 and 1. All the other terms are formed by adding the previous two terms.

a) Write a recursive method in Java that calculates the nth term.

b) Write an iterative method to solve the same problem.

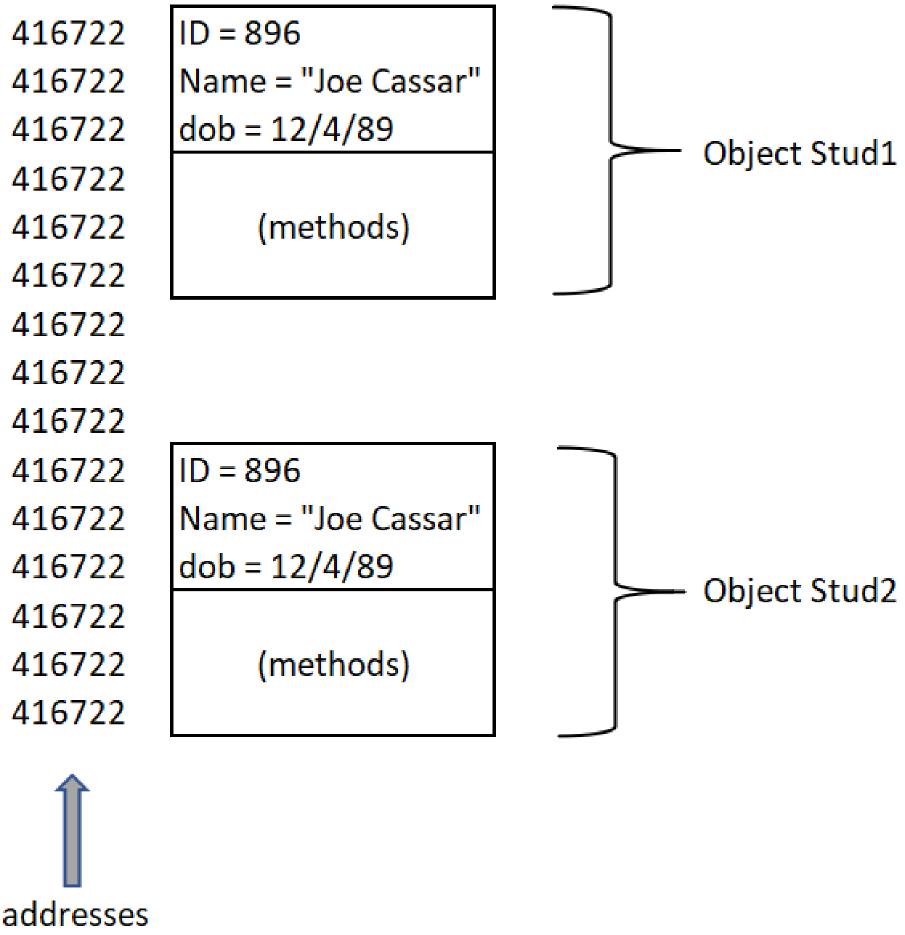
Homework 2: Primitive and Reference Types

1) Explain what primitive (atomic) types are.

2) State the eight primitive types in Java.

3) What are reference types?

4) The diagram shows two Student objects. Why does the condition (Stud1 == Stud2) is given as false and not as true.



Homework 3: Bubble Sort, Selection Sort, Linear Search and Binary Search

1) Write a class in Java that contains a 300-element one-dimensional array of type int. The class should have the following methods:

a) A method called fill1000 that fills the array with random numbers from 0 to 1000 (the array can contain repetitive values).

b) A method called viewArray that displays the elements of the array.

c) A method called bubSort that sorts the array using the bubble sort.

d) A method called selSort that sorts the array using the selection sort.

e) A method called linSearchSorted that is given an integer i and it returns the index of the first occurrence of i. If i is not found it returns -1. The linear search assumes that the array is sorted.

f) A method called linSearchNotSorted that is given an integer i and it returns the index of the first occurrence of i. If i is not found it returns -1. The linear search assumes that the array is not sorted.

g) A method called binSearch that searches for an element i using the binary search and returns an index of an occurrence of i. It returns -1 if i is not found.

h) The main method that gives repetitively these choices to the user:

   i) Fill the array
   ii) View the array
   iii) Bubble sort the array
   iv) Selection sort the array
   v) Linear search a sorted array
   vi) Linear search an unsorted array
   vii) Binary search the array
   viii) Quit program