

Question 1

(10 marks)

Write a Java class `Student` with three fields: `name`, `mark` and `maxscore` representing a student who has scored `mark` out of `maxscore`. The class has a constructor and two methods:

`addBonus` adds a given bonus score to the student's mark;

`toString` returns a `String` summarising the current state of the `Student`.

You should select suitable types and parameters for your class its constructor and methods. Javadoc comments are **not** required for this question.

Answer Question 1 here

Question 2

(10 marks)

Using your `Student` class from Question 1, write an efficient method

```
public Student topStudent(ArrayList<Student> cits1001)
```

that returns the `Student` object with the highest percentage score from a given `ArrayList` of `Students` called `cits1001`. Write a suitable helper method that returns a student's mark as a percentage rounded to the nearest integer. Javadoc comments are **not** required for this question.

Answer Question 2 here

Question 3

(10 marks)

Movie tickets at the Windsor Cinema in Nedlands are priced as follows:

Adult \$16.50

Subscriber Concession \$12.00

Senior / Pensioner / Child \$11.00

Tuesday is our cheap day at the Windsor Cinema:

\$8.50 before 6pm and \$10.50 after 6pm on the cheap days

(a) Write a signature for a method `ticketPrice` to return the cost of a movie ticket (in cents) given the day of the week, session time, and customer status (Adult, Subscriber Concession, Senior, Pensioner, Child). The method should return -1 if any of the parameters are invalid. Then design up to 5 test cases for this method, writing a JUnit `assertEquals` statement for each test case.

(b) Write code to implement the `ticketPrice` method. Include a Javadoc header for the method that describes its parameters and return value. Javadoc comments **are** required for this question.

Answer Question 3(a) here

Question 4**(10 marks)**

The tech-support project studied in labs and lectures mapped user query words to technical responses. Write a method, including a Javadoc header, with the signature

```
public String generateResponse(  
    HashSet<String> words,  
    HashMap<String, String> responseMap)
```

The method should return a selected response from its (already initialised) parameter `responseMap` linked to one of the words from the `HashSet` parameter. If no matching response is found then return "Sorry I don't know".

Answer Question 4 here

Question 5**(10 marks)**

Compare and contrast the two methods `sort1` and `sort2`. Each method sorts an array of integers. Comment on the strengths and weakness of each implementation, using the criteria of readability, efficiency, cohesion and correctness.

```
public static void sort1(int[] a)
{
    for (int pass=1; pass<a.length; pass++) {
        int x=a[pass];
        int y=pass-1;
        while (y>=0 && a[y]>x) { //ERROR pos should be y
            a[y+1]=a[y];
            y--;
        }
        a[y+1]=x;
    }
}
```

```
public static void sort2(int[] a)
{
    for (int pass=1; pass < a.length; pass++) {
        for (int j = 0; j < a.length-pass; j++) {
            if (a[j] > a[j+1]) {
                // exchange out of order elements
                swap(a, j, j+1); //SEMI was MISSING
            }
        }
    }
}
```

```
private static void swap(int[] a, int pos1, int pos2) //error needs to be static
{
    int swap = a[pos2];
    a[pos2] = a[pos1];
    a[pos1] = swap;
}
```

Answer Question 5 on the next page
