UNIVERSITY OF BOLTON
BSC COMPUTING
COURSEWORK SUBMISSION FORM

Student/Centre to complete:

SURNAME/FAMILY NAME: …………………………… FORENAMES: ……………………………….

BOLTON STUDENT ID: ……………………….. EMAIL: …..……………………………………

DATE OF SUBMISSION: …………………………………….……………

MODULE NO./TITLE:…CPU5008 Data Structures and Algorithms…………………………

TUTOR'S NAME:………Abdul Razak…………………..... ……………………

COURSEWORK TITLE: Assignment 1(part 1): Appointments: Cloning, Sorting and Searching

Please state if this is your FIRST submission OR REFERRED/DEFERRED submission
OR a REPEAT submission? □

…………………..…………………………………………………………………………

Declaration

I hereby declare that this work is my own work. I understand that if I am suspected of plagiarism or another form of cheating, my work be referred to Academic Registrar and/or the Board of Examiners, which may result in me being expelled from the programme. I understand once I submit this work, it will automatically belong to the University of Bolton.

Academic staff to complete:

Feedback: ……………………………………………………………………………………

…………………………………………………………………………………......................

……………………………………………………………………………………………………

…………………………………………………………………………………………….......

Date Issued:…..……*Week 4*………. Hand-In Date: *Week 6* **(27th October 2016 @ 12 noon)**

Other Relevant Date e.g. Demonstration **In class demonstration 27th October 2016**

Received: On Time □ Late □ (within 5 days of published deadline date)

Mark awarded: ………..% Do not apply mark penalty unless the work was submitted late.

Assessors Name: …A. Razak…….………… Signature:...

Date:…………………………

Degree Conversions	A: 70-100%	B: 60-69%	C: 50-59%	D: 40-49%	F: 0-39%
HND Conversions	Pass: 40-49%	Merit: 50-66%	Distinction: 67-100%		

Late submission

For late submission see student handbook:
http://www.bolton.ac.uk/Quality/SE/Student-Handbook/Home.aspx

Creative Technologies

Course / Programme:	**Computing**
Module name and code:	**Data Structures and Algorithms CPU5008**
Tutor:	**Abdul Razak**
Assignment Number:	**One (part 1)**
Assignment Title:	**Cloning, Sorting and Searching.**
Issue Date:	**13th October 2016**
Submission Deadline:	**27th October 2016**

Learning Outcomes:

LO 1. Implement shallow and deep copying and use the Comparable interface and Comparator class.

Assignment:

Using cloning, sorting and searching algorithms on data structures.

HE5 – Assessment is set appropriate to level HE5.

Specific Assessment Criteria

- Have solved several problem tasks using the Java language.
- Have shown the ability to decompose a problem and have designed suitable class structures to effect a solution to a given problem.
- Have demonstrated the ability to complete tasks from blank solutions in order to achieve a goal.

Grading

A percentage mark will be provided as feedback. Grading is as follows:

A:	70-100%
B:	60-69%
C:	50-59%
D:	40-49%
Marginal fail:	35-39%

Marks below 35% will be classed as fail.

Assignment 1 (part 1)

Create a new Java project called **A1_Donald_Duck** replacing ***Donald_Duck*** with your name. Use the underscore (_) to replace any space characters. In the project create the following 4 classes called Date, Time MeetingRoom and Appointment which can be used in an electronic organiser type application

Class name: Date (Attributes)	
private int day	day of the month (1-31)
private int month	month of the year (1-12)
private int year	year e.g. 2008

Class name: Time (Attributes)	
private int hour	hour in the day (0-23)
private int minute	minute within the hour (0-59)

Class name: MeetingRoom (Attributes)	
private String room	room in String format e.g. B1-10 or C2-07 or T4-32

Class name: Appointment (Attributes)	
private String name	who to meet
private String purpose	purpose of appointment
private Date date	date of appointment
private Time time	time of appointment
private MeetingRoom room	where to meet

Program Development Stage 1 (15 marks)

All four classes must have

1. a default constructor that initialises its attributes to sensible values.
2. a parameter constructor that sets all the attribute values based on the parameter values.
3. a copy constructor that performs shallow copying.
4. a setter method for each attribute.
5. a getter method for each attribute.
6. a toString method to return a suitably formatted string of the attribute values.

Create a class called **Test001** that simply creates a number of appointments and demonstrates shallow copying of Appointment objects.

Program Development Stage 2 (25 marks)

For each of the 4 classes change the class definition to implement the Cloneable interface and add appropriate methods and code to enable deep copying of Appointment objects.

Create a class called **Test002** that simply creates a number of appointments and demonstrates deep copying of Appointment objects.

Program Development Stage 3 (30 marks)

For each of the 4 classes change the class definition to implement the Comparable interface and add appropriate methods and code to enable natural ordering (chronological order) of Appointment objects. (Hint. One way would be to construct a `Long` integer that encompasses date and time, then sort according to this)

Create a class called **Test003** that simply creates a number of appointments and stores them in

(i) An ArrayList of Appointment objects
(ii) An array of Appointment objects (same objects, in the same original order as the ArrayList)

(a) Create an Appointment reference called searchAppointment and assign it a reference to an unsorted Appointment object.
(b) Create another Appointment reference called searchAppointmentClone and assign it the clone of searchAppointment.

Now demonstrate sorting of Appointment objects in both array and ArrayList. Note that you must use the for-each loop with the ArrayList.

All the ArrayLists and interfaces must be type safe using the parameterised type declaration.

Q. Do you have a problem with sorting the array if the array dimension is declared greater than the number of appointments? Why might this be?

Now try

```
Arrays.sort(appArray, 0, count);
```

where appArray is the array and count is the last filled index of the array.

This is one of several reasons why it is better in general to prefer an ArrayList to an array.

After the sorting, search for the objects of searchAppointment and searchAppointmentClone in both the array and Arraylist. Demonstrate that they have been found.

Q. Does the search look for the same object references or the same attribute values?

Q. What exactly does the search try to match up? Can you verify your answer?

Program Development Stage 4 (30 marks)

Implement the Comparator interface for the Appointment class and create a Comparator class called `SortByRoomFloor`.

Create a class called **Test004** that now re-sorts both array and ArrayList from stage 3 according to the floor of the meeting room (lowest first). If some rooms are on the same floor, then these rooms should be ordered naturally, i.e. alphabetically. Produce suitable console output to demonstrate the code.

Q. Does this test still work if the Comparable interface is removed?

Q. Would you say that the Comparator class is a better option for this stage 3 than the Comparable interface? If so, why?

After the sorting, search for the objects of searchAppointment and searchAppointmentClone in both the array and Arraylist. Demonstrate that they have been found.

Q. Does the search based on natural ordering still work? If not why not? What does work?

IMPORTANT
At all stages in the development you should add JavaDoc comments to explain the purpose of the new methods added to the each class. In addition to the JavaDoc comments you should add additional normal style comments where appropriate.

Note: In the Eclipse IDE normal comments appear in green and JavaDoc comments appears in blue.

What you must submit

Your must submit the assignment in class by 12.00 noon on the Tuesday of the week shown on page 1 of this assignment brief.

Your submission must include:

1) This assignment brief completed and signed on the front page, also complete the table below.
2) Your answers (and discussion if appropriate) to the questions highlighted in yellow.
3) Printouts of the code for the classes
    - `Appointment`, `Date`, `Time`, `MeetingRoom`
    - `Test001` to `Test004`
4) The entire Eclipse project on a CD (copy the entire project folder from the Eclipse workspace but **do not** change the folder name afterwards)
5) The CD should be clearly identified with your name, student ID, module name and assignment title and secured with the above documentation

Failure to submit all the above will result in a loss of marks

In this assignment I have achieved the following objectives			
Tick appropriate box NA – not attempted : Part – part completed : Full – fully completed	**NA**	**Part**	**Full**
Stage 1			
Stage 2			
Stage 3			
Stage 4			

Assessment Criteria

The mark you get is based on

- the stage of development you have achieved
- the quality of the code provided
  - well structured
  - good use of identifier names
  - good use of comments (both JavaDoc and normal comments)
- clear concise algorithms that uses appropriate Java programming constructs
- Answers to the questions

NOTE: You need to demonstrate your programs and explain how they work during the practical session on Tuesday 27th October 2016.

No demonstration means zero mark.