THE UNIVERSITY OF HUDDERSFIELD

School of Computing and Engineering

ASSIGNMENT SPECIFICATION

Module	Details
Module Code	CMS3506
Module Title	Software Development
Course Title/s	MSc Information Systems Management MSc Computing

Assessment	Weighting, Type and Contact Details
Title	Coursework: Object-Oriented Programming
Weighting	50%
Mode of working for assessment task	Individual Note : if the assessment task is to be completed on an individual basis there should be no collusion or collaboration whilst working on and subsequently submitting this assignment.
Module Leader	Ilias Tachmazidis
Module Tutor/s	

Submission	Submission and Feedback Details
Hand-out date	16 January 2023
How to submit your work.	Via Brightspace (to be automatically checked by TurnItIn)
Submission date/s and times	Final submission is 06-02-2023 12:00
Expected amount of independent time you should allocate to complete this assessment	15 hours

Submission	Submission and Feedback Details
Submission type and format	A 'zip' file (or similar) containing the software developed and an accompanying report. Details below.
Date by which your grade and feedback will be returned	27 February 2023

Additional Guidance Information	Details
Your responsibility	It is your responsibility to read and understand the University regulations regarding conduct in assessment. Please pay special attention to the assessment regulations (section 10) on Academic Misconduct. In brief: ensure that you; 1. DO NOT use the work of another student - this includes students from previous years and other institutions, as well as current students on the module. 2. DO NOT make your work available or leave insecure, for other students to view or use. 3. Any examples provided by the module tutor should be appropriately referenced, as should examples from external sources. Further guidance can be found in the SCEN Academic Skills Resource and UoH Academic Integrity Resource module in Brightspace. If you experience difficulties with this assessment or with time management, please speak to the module tutor/s, your Personal Academic Tutor, or the School's Guidance Team. (sce.guidance@hud.ac.uk).
Requesting a Late	It is expected that you complete your assessments by the published deadlines. However, it is recognised that there can be unexpected

Additional Guidance Information	Details
Submission	circumstances which may affect you being able to do so. In such circumstances, you may submit a request for an extension. Extension applications must be submitted before the published assessment deadline has passed. There are two types of extension that you may request. You will be required to indicate which one you are applying for when you submit the request for Late Submission via MyHud/MyStudies. 1. Self-certified illness extension of up to 5 working days. • Evidence will not be required for this type of request, but you are limited to two self-certified extension requests in any academic year. 2. Extension request of up to 10 working days. • This extension requires you to submit appropriate evidence in support of your request. The maximum extension that can grant is 10 working days. <u>Accepted grounds for an extension</u> • Serious short-term illness or accident (of a nature which in employment would result in a health-related absence); • Evidence of a long-term health condition worsening; Emerging mental health condition, or worsening of an existing mental health condition; • Bereavement. If you are unable to submit work within the maximum late submission period of 10 days, contact the School's Guidance Team. (sce.guidance@hud.ac.uk), as you may need to submit a claim for Extenuating Circumstances (ECs).

Additional Guidance Information	Details
Extenuating Circumstances (ECs)	An EC claim is appropriate in exceptional circumstances, when an extension is not sufficient due to the nature of the request. You can access the EC claim form on the Registry website; where you can also find out more about the process. You will need to submit independent, verifiable evidence for your claim to be considered. Once your EC claim has been reviewed you will get an EC outcome email from Registry. If you are unsure what it means or what you need to do next, please speak to the Student Support Office – Room SJ1/01 An approved EC will extend the submission date to the next assessment period (e.g July resit period).
Late Submission (No ECs approved)	Late submission, up to 5 working days, of the assessment submission deadline, will result in your grade being capped to a maximum of a pass mark. Submission after this period, without an approved extension, will result in a 0% grade for this assessment component.
Tutor Referral available	YES
Resources	• Please note: you can access free Office365 software and you have 1 Tb of free storage space available on Microsoft's OneDrive – Guidance on downloading Office 365.

Coursework: Object-Oriented Programming

1. Assignment Aims

To develop skills in object-oriented programming.

To develop skills in writing simple programs.

To demonstrate the need to document software.

2. Learning Outcomes:

Knowledge and Understanding Outcomes

1. Appraise the requirements for an information system.

2. Analyse the design requirements for a software system using an object-oriented approach.

3. Evaluate the available programming languages for software systems development.

3. Assessment Brief

You are required to design, implement, test and document **Java programs for each activity** as described below.

Formative Feedback: Opportunity for formative assessment will be **provided during practical sessions**. Work will be set and feedback will be provided on the students' efforts. Students will be able to gain **informal feedback on work in progress by taking it to the practical for their tutor to review**.

Final Submission: By the deadline of **06-02-2023 12:00**, you must submit a **'zip' file** (or similar) containing:

1. An **electronic report** (preferably in **PDF format**) that includes **your answers to each activity**. More specifically, for each sub-question you should provide **screenshots of your code and its corresponding output** to demonstrate that you have solved the problem as stated in the activity. For your own program (i.e. 'A Program of your Own'), you should include a **walk-through**, namely screenshots and explanations of your program. **All activities should have an explicit indication of all relevant classes**.

2. An **electronic copy of your classes** (Java code relevant to all activities). Electronic copies should be submitted within a **single IntelliJ project**. All code should be professionally written (including comments and indentation, use of sensible variable names, etc.).

Note: Please make sure that your **report includes the list of relevant classes** (of your IntelliJ project) for each activity (where applicable). In addition, **your IntelliJ project must work on tutor's machine as submitted** (this is guaranteed as long as you follow taught material, **do not use external libraries besides basic Java**).

How and where to hand in: Please create a folder labelled with your student ID and name (e.g.: LastName_FirstName_u1234567). **Place your report in this folder with your IntelliJ project**. **Compress the folder** using standard compression software and **submit the compressed file** (preferably a 'zip' file) to Brightspace. There will be a link for this under "Assessment". You may upload as many times as required; **the latest upload (on time)** will be the only one that **will be considered for marking**.

Tutor Reassessment: Any student who scores **0-49%** in this assessment will be informed when the marked work is returned (27 February 2023). A period of three weeks will be allowed for students to submit their **Tutor Reassessment** (**due on 20/03/2023 12:00**). Submission of Tutor Reassessment is via Brightspace, there will be a link for this under "Assessment". The Tutor Reassessment will be marked and the marks returned within three weeks (02 May 2023).

All activities (i.e. Activities 1-7) for this coursework are described below.

Activity 1: Statements and Sequences [5 X 3 = 15%]

Write Java programs to do the following:

a) Convert Pounds to Dollars, where 1 GBP is equal in value to 1.2 USD
b) Convert Fahrenheit to Celsius, by subtracting 32 and then dividing by 1.8
c) Convert Celsius to Fahrenheit by multiplying by 1.8 and then adding 32
d) Write a Java program that takes four numbers as input then calculates and displays their average
e) Take N% discount off a price P

Activity 2: Conditions and Conditional Statements [5 X 3 = 15%]

Write a sequence of statements involving an If-then structure, and printing out a message:

a) To find whether a given year is a leap year or not
b) To test if the date entered comes before or after today (use the LocalDateTime class to handle dates)
c) To read the age of a candidate and determine whether they are eligible to vote
d) To find the largest of three numbers
e) To convert a month number (1 to 12) entered by the user into the month name

Activity 3: Iteration [5 X 3 = 15%]

Write Java programs to do the following:

a) Use type char (e.g. (char) 65 = 'A') to print the alphabet in both lower case (a-z) and upper case (A-Z)
b) Print the alphabet backwards (both lower and upper case)
c) Print every second letter in the alphabet (both lower and upper case)
d) Ask the user to enter a number – find how many times the number can be halved before it becomes smaller than 1
e) Generate a random number between 1 and 100. Ask the user to guess, and prompt with “Too High” or “Too Low” until they guess it correctly

Activity 4: Arrays [5 X 3 = 15%]

Write Java programs to do the following:

a) To store elements in an array (5 numbers given as input) and print the array
b) To display the contents of an array (5 numbers given as input) in reverse order
c) To find the sum of all elements in an array
d) To separate odd and even integers into separate arrays
e) Search for an element in a 2D array

Activity 5: Functions [5 X 3 = 15%]

Write Java programs to do the following (make sure the parameters are defined with the most appropriate data types):

a) Ask the user to enter their full name, and then call a function that will display the message “Hello” followed by the provided name
b) Ask the user to enter two numbers, and then call a function in which the product of these two numbers is returned. Display the returned value in your main program
c) Write a function to ask the user for a Price and a VAT rate (in %) and then return the resulting price including VAT. Display the returned value in your main program
d) Ask the user to enter the time of day in three parts – hours, minutes and seconds, and then call a function that calculates and returns the number of seconds since midnight last night. Display the returned value in your main program
e) Write a function to check if a word specified by the user is a palindrome. Display the returned value in your main program

Activity 6: Classes [5 X 3 = 15%]

Write Java programs to do the following:

a) Provide code for creating a Rectangle class in Java. The class should have attributes for the height and width of the rectangle and one method named “Area”
b) Provide code for creating a Triangle class in Java. The class should have attributes for the height and width of the triangle and one method named “Area”
c) Provide code for creating a Circle class in Java. The class should have an attribute for the radius of the circle and methods named “Area”, “Diameter” and “Circumference”
d) Provide code for creating a Square class in Java. Class Square should be a subclass of class Rectangle and should check that height and width are equal
e) Provide code to create new instances of Rectangle, Triangle, Circle and Square, give them names and print the returned values of their methods

Activity 7: A Program of your Own [10%]

You should write a Java program that solves an interesting problem (a problem that was not already solved in class). The program can do anything you like provided it includes all taught components, namely:

1) Statements and Sequences
2) Conditions and Conditional Statements
3) Iteration
4) Arrays
5) Functions
6) Classes

4. Marking Scheme

Each sub-question in **Activities 1-6** is worth **3%** as indicated above. **Marks will be awarded** for each sub-question **as follows**:

Correct (3 points):

- Complete implementation
- Correct implementation
- Clear and well written code
- Implementation included in electronic report and IntelliJ project

Minor Errors (2 points):

- Complete implementation
- Minor errors in implementation
- Clear and well written code

Major Errors (1 points):

- Some components implemented
- Major errors in implementation
- Code is easy to read

Incorrect/Missing (0 points)

- Missing
- Incorrect
- Does not follow specifications

Note: Activities 1-6 cover specific taught components, and thus, **marks will be awarded only if the taught components are present** (e.g. Activity 5 requires **function calls**, Activity 6 requires **separate classes**, etc.).

Activity 7 (i.e. 'A Program of your Own') will be **evaluated as follows**:

Correct (10 points):

- Correct implementation
- All taught components work as expected
- Clear and well written code
- Comments describing various parts of the code are included
- Implementation included in electronic report and IntelliJ project

Minor Errors (8 points):

- Minor errors in implementation
- At least 5 taught components implemented
- Clear and well written code
- Comments describing various parts of the code are included
- Implementation included in electronic report and IntelliJ project

Major Errors (6 points):

- Major errors in implementation
- Most taught components implemented
- Code is easy to read
- Comments are included

Borderline (4 points):

- Some taught components implemented
- Incomplete implementation

Poor Attempt (2 points):

- Poor or little code
- Implementation shows little evidence of progress

Incorrect/Missing (0 points)

- Missing
- Incorrect
- Does not follow specifications

5. Grading Rubric

Marks will be calculated as the sum of awarded points (as described above).

Postgraduate/Masters-Level Modules

	Pass/Referral Bands
Pass	50%
Refer	0-49%
	Module Grading Bands
A	70% +
B	60-69%
C	50-59%
R**	0-49%
F**	0-49%

Postgraduate/Masters-Level Modules**
Initial CAB consideration allows a referral opportunity between 0% and 49%
CAB consideration following an initial will conclude fail between 0 and 49%
R = Referral
F = Fail