ASSESSMENT BRIEF

COURSE: Bachelor of Information Technology/ bachelor of Information Technology with a specialisation in Cyber Security

Unit:	Object Oriented Design and Programming
Unit Code:	OODP101
Type of Assessment:	Assessment 3 – Solution to programming problem by group of 3-4 students
Length/Duration:	20 Hours
Unit Learning Outcomes addressed:	Upon successful completion of this unit students should be able to: 1. Demonstrate basic knowledge of object-oriented programming concepts and programming problems 2. Analyse and dissect simple design and programming problem 3. Implement a well-designed modularized solution to small programming problems 4. Develop and/or implement testing schedules
Submission Date:	Week 10 (Assessment on moodle) and week 11(demonstration in class)
Assessment Task:	A group of 3-4 students will work together to provide a quality solution to programming problem using JAVA programming language,
Total Mark:	Assignment and Demonstration 30 Marks
Weighting:	30% of the unit total marks

Students are advised that **submission of an Assessment Task past the due date without a formally signed approved Assignment Extension Form** (Kent Website *MyKent Student Link> FORM – Assignment Extension Application Form – Student Login Required)* **or previously approved application for other extenuating circumstances impacting course of study, incurs a 5% penalty per calendar day**, calculated by deduction from the total mark.
For example. An Assessment Task marked out of 40 will incur a 2 mark penalty for each calendar day.

More information, please refer to *(*Kent Website *MyKent Student Link> POLICY – Assessment Policy & Procedures – Student Login Required)*

ASSESSMENT DESCRIPTION:

In this assessment, student will have to pick one scenario from the following or use their creative thinking and choose their own scenario similar to below mentioned so that they can demonstrate the principles of Object Oriented Programming (Creation of classes and objects, Encapsulation, Inheritance and Polymorphism) and fundamentals of programming (Structures of programming, array, Modularisation). Students are required to demonstrate the work done by them during week 12 tutorial to get marks for demonstration task mentioned in marking guide.

Note for Tutors: - Please approve the scenarios of each team and make sure that each scenario is picked and not repeated by too many teams. If students have designed their own scenario, make sure it has parent child relationship and main principles of OOP are demonstrated.

Scenario 1	You are working in one company and employer wants to create an application to store all employee's information and calculate amount paid to employees in each week, month and year. Company is having three types of employees which are full time, part time and casual employee. Company provides recreation activities each month to their permanent employees. **Step 1:-** Create one employee class as parent class with basic attributes that you think can be part of employee class and company wants to store information about and that can be used to calculate the pay of employee. Write one default and one parametrized constructor, getter, setter and override toString() method to display the object's attributes, another method to calculate the pay of employee. **Step 2:-** Create child classes full time, part time, casuals. All child classes will inherit basic attributes from parent but will have their own attributes as follows: **Full time:-** annual leave hours, sick leave hours, recreation amount which will be $100 deducted from employee pay **Part time**:- number of days, recreation amount which will be $50 deducted from employee pay **Casuals:-** casual loadings which is 10% of the pay added to final pay So, in child classes, write one default and one parametrized constructor, getter, setter and override toString() method and a method to calculate the pay will be overridden with above mentioned detail detail. **Step 3:-** Create one driver class which will have main method and will do following 1. Create five objects of each type of employee with different and meaningful details. 2. Use skills that you learnt in fundamentals of programming and print the employee details in readable format and amount of money paid by company to each type of employees in one week, one month and one year. (Be creative to do this task and displaying the details). While creating classes, make sure you create all different classes and demonstrate the principles of OOP and provide comment in code to illustrate it. Clearly show where have you created classes, used polymorphism, inheritance and

	encapsulation. **Step 4:-** Create one class diagram to illustrate your classes and their relationships. **Step 5**:- Test your program and if it runs successfully then provide screenshots in word document. **Step 6:-** Be available in week 11 class to demonstrate your code. Your tutor will ask you questions from your assessment part that you have worked on, and you will get marks according to your answers.
Scenario 2	You have opened a small dairy shop and have started to stock three types of products which are dairy, bakery and Fresh. You are creating this application for inventory management and you want to display all details of products and find out total amount paid to supplier from this application. **Step 1:-** Create one Product class as parent class with basic attributes like barcode, price, name, expiry date and others that you think can be part of product class and company wants to store information about and that can be used to find the items that are due to expire at certain day. Write one default and one parametrized constructor, getter, setter and override toString() method to display the object's attributes, setter method to find out the expiry date of each product should be created but that will be overridden by children classes. **Step 2:-** Create child classes Fresh, Dairy, Bakery. All child classes will inherit basic attributes from parent but will have their own attributes as follows: **Fresh:-** type(fillets, sausages etc), meatOf (values can be chicken, lamb, goat etc) , use by date that is 4 days after the packaging date. **Dairy:-** produceOf(cow milk, goat milk etc), type (cheese, butter , milk , yogurt etc), expiry date that is 8 days after the packaging date **Bakery-** itemtype (bread, dinner roll, hamburger buns etc), baked on date, best before date that is 3 days after the bakes on date. So, in child classes, write one default and one parametrized constructor, getter, setter and override toString() method and a method to find the expiry date of product should be overridden with necessary details. **Step 3:-** Create one driver class which will have main method and will do following 1. Create five objects of each type of product with different and meaningful details.(Take hints from example) 2. Use skills that you learnt in fundamentals of programming and print the product details in readable format and details of product expiring on certain day of your choice. (Be creative to do this task and displaying the details). While creating classes, make sure you create all different classes and demonstrate the principles of OOP and provide comment in code to illustrate it. Clearly show where have you created classes, used polymorphism, inheritance and encapsulation.

	Step 4:- Create one class diagram to illustrate your classes and their relationships. **Step 5**:- Test your program and if it runs successfully then provide screenshots in word document. **Step 6:**- Be available in week 11 class to demonstrate your code. Your tutor will ask you questions from your assessment part that you have worked on, and you will get marks according to your answers.
Scenario 3	You are working as a security officer at a college campus and building manager has asked you to prepare a report on number of people visiting the campus on particular day and whether they are admins, teaching staff or student and their details. You are required to create this application using java programming language. There are three type of people which admins, teaching staff and student. **Step 1:-** Create one People class as parent class with basic attributes like name, ID, phone number, day attending the campus and others that you think can be part of people class and company wants to store information about and that can be used to display the people attending the campus on certain day. Write one default and one parametrized constructor, getter, setter and override toString() method to display the object's attributes, a setter method for days attending would be provided and will be overridden by all children classes. **Step 2:-** Create child classes Admins, Teaching Staff, Student. All child classes will inherit basic attributes from parent but will have their own attributes as follows: **Admins**:- responsibilities, department, days attending would be depending upon the department. Enrolment department are attending only Monday and Tuesday. Higher education department attending Monday to Thursday, reception department is attending Monday to Friday. So override days attending setter method to use above details. **Student:-** course, number of units, days attending would be depending on course, BIT students are attending campus on Monday , Tuesday, BACT students are attending on Wednesday , thursday and BBUS students are attending on Friday. **Teaching Staff:-** type, days attending would be depending upon the type of staff, full time attending campus from Monday to Friday, part time attending on Thursday and Friday So, in child classes, write one default and one parametrized constructor, getter, setter and override toString() method and a setter method for days attending would overridden with above details. **Step 3:-** Create one driver class which will have main method and will do following 1. Create five objects of each type of People with different and meaningful details.(Take hints from explanation) 2. Use skills that you learnt in fundamentals of programming and print the People details in readable format and details of People attending campus

	on certain day entered by users. (Be creative to do this task and displaying the details). While creating classes, make sure you create all different classes and demonstrate the principles of OOP and provide comment in code to illustrate it. Clearly show where have you created classes, used polymorphism, inheritance and encapsulation. **Step 4:-** Create one class diagram to illustrate your classes and their relationships. **Step 5**:- Test your program and if it runs successfully then provide screenshots in word document. **Step 6:-** Be available in week 11 class to demonstrate your code. Your tutor will ask you questions from your assessment part that you have worked on, and you will get marks according to your answers.

ASSESSMENT SUBMISSION:

Group leader will submit java project (coding), a word file having class diagram and testing screenshots and individual contribution statement. All these should be put inside a zip/compressed file for submission. Java project (coding) should be exported from eclipse.
All other members will submit their own individual contribution statement.

This assignment must be submitted online in Moodle.

For assistance please speak to our Academic Learning Skills Coordinators, in Sydney (als.syd@kent.edu.au) or in Melbourne (als.mel@kent.edu.au). They can help you with understanding the task, draft checking, structure, referencing and other assignment-related matters.

GENERAL NOTES FOR ASSESSMENT TASKS

Content for Assessment Task papers should incorporate a formal introduction, main points and conclusion.

Appropriate academic writing and referencing are inevitable academic skills that you must develop and demonstrate in work being presented for assessment. The content of high quality work presented by a student must be fully referenced within-text citations and a Reference List at the end. Kent strongly recommends you refer to the Academic Learning Support Workshop materials available on the Kent Learning Management System (Moodle). For details please click the link http://moodle.kent.edu.au/kentmoodle/mod/folder/view.php?id=3606 and download the file titled "Harvard Referencing Workbook". This Moodle Site is the location for Workbooks and information that are presented to Kent Students in the ALS Workshops conducted at the beginning of each Trimester.

Kent recommends a minimum of **FIVE (5)** references in work being presented for assessment. Unless otherwise specifically instructed by your Lecturer or as detailed in the Unit Outline for the specific Assessment Task, any paper with less than five (5) references may be deemed not meeting a satisfactory standard and possibly be failed.

Content in Assessment tasks that includes sources that are not properly referenced according to the "*Harvard Referencing Workbook*" will be penalised.

Marks will be deducted for failure to adhere to the word count if this is specifically stated for the Assessment Task in the Unit Outline. As a general rule there is an allowable discretionary variance to the word count in that it is generally accepted that a student may go over or under by 10% than the stated length.
Students are not allowed to use AI tools for the preparation of their submissions.

GENERAL NOTES FOR REFERENCING

References are assessed for their quality. Students should draw on quality academic sources, such as books, chapters from edited books, journals etc. The textbook for the Unit of study can be used as a reference, but not the Lecturer Notes. The Assessor will want to see evidence that a student is capable of conducting their own research. Also, in order to help Assessors determine a student's understanding of the work they cite, all in-text references (not just direct quotes) must include the specific page number(s) if shown in the original. Before preparing your Assessment Task or own contribution, please review this 'YouTube' video (Avoiding Plagiarism through Referencing) by clicking on the following link: link: http://moodle.kent.edu.au/kentmoodle/mod/folder/view.php?id=3606

A search for peer-reviewed journal articles may also assist students. These type of journal articles can be located in the online journal databases and can be accessed from the Kent Library homepage. Wikipedia, online dictionaries and online encyclopaedias are acceptable as a starting point to gain knowledge about a topic, but should not be over-used – these should constitute no more than 10% of your total list of references/sources. Additional information and literature can be used where these are produced by legitimate sources, such as government departments, research institutes such as the National Health and Medical Research Council (NHMRC), or international organisations such as the World Health Organisation (WHO). Legitimate organisations and government departments produce peer reviewed reports and articles and are therefore very useful and mostly very current. The content of the following link explains why it is not acceptable to use non-peer reviewed websites (Why can't I just Google?): https://www.youtube.com/watch?v=N39mnu1Pkgw (Thank you to La Trobe University for access to this video).

MARKING GUIDE (RUBRIC):

Your answers for the final examination questions will be assessed as per the following marking criteria. Please read carefully each section/level and marks weightage.

Criteria	High Distinction (85 – 100%)	Distinction (75 – 84%)	Credit (65 -74%)	Pass (50 – 64%)	Fail (0 – 49%)
Delivery (30 Points)	Completed between 95-100% of the requirements. Submitted in correct format.	Completed between 85-95% of the requirements. Submitted in correct format.	Completed between 75 – 85% of the requirements. Submitted in correct format.	Completed between 50 – 75% of the requirements. Submitted in correct format.	No task submitted. Completed less than 50% of the requirements. Not submitted in correct format.
Implementation (15 Points)	The program compiles without errors. Proper comments to explain each block of code, proper indentation, proper code layout and meaningful variable names. Code is logical and appropriate. Creatively organised work. Excellent use of variables (no global variables, unambiguous naming).	The program compiles without errors. Not enough comments to explain each block of code, proper indentation, proper code layout and meaningful variable names. Code is logical and appropriate. Organised work. Good use of variables (no global variables, unambiguous naming).	The program compiles without errors. Not enough comments, no proper layout of code and no proper indentation but variable names are proper. Code is logical and appropriate. Organised work. Good use of variables (few global variables, unambiguous naming).	The program compiles with errors. Major issues in indentation, layout and variable names. Code is not logical and appropriate. Organised work. Good use of variables (many global variables, ambiguous naming).	No attempt made. Poor use of white space (indentation, blank lines). Disorganised work. Poor use of variables (many global variables, ambiguous naming).
Efficiency (5 Points)	Solution is efficient, easy to understand, and maintain.	Solution is efficient and easy to follow.	A logical solution that is easy to follow but it is not the most efficient.	A solution which provides some answer but not efficient	A difficult and inefficient solution.
Testing (10 Points)	Proper testing screenshots are attached, and tests identified clearly	Screenshots for only a few tests are provided but they can be identified clearly	All screenshots are provided but it is difficult to identify tests clearly	Only a few outputs are displayed but no relation to tests	No attempt made
Runtime (10 Points)	Executes without errors, excellent user prompts, good use of symbols, spacing in output. Thorough and organised testing has been completed and output from test cases are included.	Executes without errors. User prompts are understandable, minimum use of symbols or spacing in output. Thorough testing has been completed.	Executes without errors. User prompts contain little information, poor design, some testing has been completed.	Executes with some warning errors.	Does not execute due to errors. User prompts are misleading or non-existent. No testing has been completed.

Documentation and Demonstration (20 Points)	Clearly and effectively documented including descriptions of all variables. specific purpose is noted for each function, control structure, input requirements and output results. Demonstrated high ethical standards and honesty in delivery.	Clearly documented including descriptions of all variables. Specific purpose is noted for each function, control structure. Demonstrated sufficient ethical standards and honesty in delivery.	Basic documentation has been completed including descriptions of all variables. purpose is noted for each function. Demonstrated moderate ethical standards and honesty in delivery.	Basic documentation has been completed. Demonstrated adequate ethical standards and honesty in delivery.	No attempt made. Failed to demonstrate any ethical standards and honesty in delivery.
Teamwork & Group cohesiveness (10 Points)	Team formed, all team works has been distributed and the team members are aware of all their tasks and participation	Team formed, almost all team works has been distributed and the team members are aware of all their tasks and participation	Team formed, some team works has been distributed and the team members are aware of some of their tasks and participation	Team formed, less distribution of team works, and the team members are aware of few of their tasks and participation	No Team formed or No team works has been distributed and the team members are not aware of their tasks and participation