*NOTES ABOUT THE USE OF THIS FORM:*

1. *This form is designed to be completed on a computer. Cells in the table below will expand to accommodate any amount of text … but we suggest that you keep the narrative as succinct as possible!*
2. *Please keep the use of formatting to a minimum. Importing formatted text onto a virtual learning platform presents challenges!*
3. *This form assumes that the “unit of learning” is a module. The module, in turn, would be included in a “course” (which is not referred to here). Each module will have a series of components which have been called “units” – they may be called something different in your design (like “weeks”, or “sections”) and you are free to change the terminology.*
4. *In the section about the authors of and contributors to the course, we have provided space for 5 co-authors (or co-contributors). If there were more than six people on the team, please add additional rows to the table.*
5. *Please ensure that you use student-friendly language. So the intended learning outcomes will be framed using the word “you”, and not “the student”. (This may be at odds with what you understand to be “academic” language. The aim, in online and blended learning, is to use language that includes the student to the greatest extent possible.)*
6. *Please note that module-level outcomes should be “overarching” outcomes onto which the unit-level outcomes map. You should have a few (maybe 4) module-level outcomes, and a very few (two or three at the most) unit-level outcomes for each unit.*
7. *The unit-level template should be copied so that there is a copy of the template for EACH unit/week/section. Thus, if there are 15 units/weeks/sections in a module, you will copy the template 14 times and complete each copy for one unit/week/section.*
8. *In the unit-level template, there is a space for a detailed description of student and teacher engagement with the unit. Here we would expect to see a “blow-by-blow” account of how the unit “hangs together”. What happens first? And then? What resources would students need to access for each part of the unit’s work? Where would they find these? Where is collaboration expected to happen? How is it scaffolded? And so on? What happens in class? What happens online? How do these elements build on each other? How long should students spend on each part of the unit?*

*This is NOT a list of things that students (or teachers) do. It is a* ***detailed description*** *of the* ***process****.*

*We have used a generic set of headings in the template. You are free to change the headings to suit the particular unit, but you are* ***not*** *free to ignore any of the required information.*

*Be sure, when completing the unit-level template to contextualise the content … by which we mean that content needs to be grounded in real life – even mathematical equations need to be demonstrably linked to real life! A student needs to know* ***why*** *they are engaging with the content.*

There are 2 templates on the following pages. The **Module-level template**should be completed once, and the **Unit-level template** should be completed in respect of each of the Units (or Sections, or Weeks) in the Module

MODULE-LEVEL TEMPLATE

|  |  |
| --- | --- |
| **Details of institution that has developed the module** | |
| Name of University | University of Energy and Natural Resources (UENR) |
| Name of institutional contact | Prof. Adebayo Felix Adekoya |
| Email address of institutional contact | adebayo.adekoya@uenr.edu.gh |

|  |  |
| --- | --- |
| **Details of Creative Commons licence** (<https://creativecommons.org/licenses/>) | |
| Licence type | This module of Data Structures and Algorithms (DSA) and related material is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Licence](https://creativecommons.org/licenses/by-nc-sa/4.0/) |

|  |  |
| --- | --- |
| **Details of the authors of/contributors to the course and their role** *(You can delete any sections that don’t apply.)* | |
| Original author (if applicable) | **N/A** |
| Lead author (+ email address) | **Dr. Benjamin Asubam Weyori (benjamin.weyori@uenr.edu.gh)** |
| *Responsible for:* | **Providing guidelines on Blended Learning course development and facilitating the course development** |
| Co-author/co-contributor | **Ms. Faiza Umar Bawah** |
| *Responsible for:* | **Course development** |
| Co-author/co-contributor | **Dr. Isaac Kofi Nti** |
| *Responsible for:* | **Course development** |
| Co-author/co-contributor | **Dr. Samuel Boateng** |
| *Responsible for:* | **Course development** |
| Co-author/co-contributor | **Mr. Nicodemus Songose Awarayi** |
| *Responsible for:* | **Providing the outline of the existing course for the redevelopment into Blended Learning Module and providing useful learning resources.** |
| Co-author/co-contributor | **Dr. Owusu Nyarko-Boateng** |
| *Responsible for:* | **Technical advisor to the course development team** |

|  |  |
| --- | --- |
| **Information regarding format of material to upload onto the OER Africa repository** | |
| Primary resource (Not PDF) | MS Word Format |
| Will a Moodle common cartridge be uploaded as well? | No |

*(A Moodle common cartridge is a .ZIP file of your module – if it is created in Moodle – that can be imported into another university’s Moodle platform.)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Course details** | | | |
| Module title: | Data Structures and Algorithms | | |
| Under- or Postgraduate? | Undergraduate | Year of study: | 2 |
| Class contact time (hours): | 4 | Number of credits: | 3 |
| Private/online study hours: | 2 | Number of weeks of study: | 14 |
| Total student learning hours: | 6 | Number of units of study: | 11 |

|  |  |
| --- | --- |
| Programme(s) which might include this Module: | Bachelor of Science in Computer Science, Bachelor of Science in Information Technology, Bachelor of Science in Computer Engineering, and other related areas. |
| Pre-requisite student abilities and knowledge: | Basic to intermediate knowledge of C++/Java concepts and ability to write and execute program codes in C++/Java programming language |
| Pre-requisite (or co-requisite) modules: | Object-Oriented Programming in C++/Java |

|  |  |
| --- | --- |
| Aim of the module: | This module has been designed to equip you with first-hand experience in writing algorithms and a systematic approach to solving real-world problems with the help of fundamental data structures using object-oriented programming paradigms. |
| Brief description of module: | Two of the most critical aspects of Computer Science are data structure and algorithms. They are considered the core or the root and foundation of Computer Science. While data structures enable us to organise and store data, algorithms enable us to process it meaningfully. When you learn about Data Structures and Algorithms (DSA), you will gain a better understanding of how to employ various solutions to familiar challenges. It also teaches you how to evaluate an algorithm's performance. This allows you to select the best option from a variety of options. |

|  |  |
| --- | --- |
| Intended learning outcomes: | *At the end of this* ***module****, you will be able to:*   1. describe the fundamental concept and theories of linear and non-linear data structures. 2. analyse the algorithmic complexity of simple, non-recursive programs and explain acceptable data structures and algorithms for solving real-world challenges 3. apply effective and novel algorithmic solutions to small-scale programming problems involving data structures and recursion, both independently and collectively |
| Indicative content: | **Week 1:** Review of C++ & Introduction to Data Structures and Algorithms  **Week 2:** Stacks  **Week 3:** Queues  **Week 4:** Arrays and Linked Lists  **Week 5:** Hashing technique and Linear probing  **Week 6:** Recursion  **Week 7:** Mid Semester Examinations  **Week 8:** Sorting Algorithms  **Week 9:** Searching Algorithms  **Week 10:** Trees I  **Week 11:** Trees II  **Week 12:** Graphs  **Week 13 & 14:** Revision andEnd of Semester Examinations |
| Form of final/summative assessment: | Continuous Assessment  Attendance - 5%  Assignments - 10%  Research paper/presentation - 10%  Mid-Semester Examination - 15%  End of Semester Examination - 60%  \*If you attend 75% or more of the lectures (face-to-face and online) you will get the full mark for attendance. However, you will get half of the mark if your attendance falls below 75%. |

|  |  |
| --- | --- |
| **Assessment of module-level learning outcomes** | |
| Module-level learning outcome | Module assessment task |
| 1. Describe the fundamental concept and theories of linear and non-linear Data Structures | You will be assessed using assignments, quizzes (online and face-to-face), individual projects, group presentations, case studies, attendance, class participation, and examinations (i.e., fixed time examinations). |
| 1. Analyse the algorithmic complexity of simple, non-recursive programs and explain acceptable data structures and algorithms for solving real-world challenges |
| 1. Apply effective and novel algorithmic solutions to small-scale programming problems involving data structures and recursion, both independently and collectively |

|  |
| --- |
| **Significant features or elements of module** |
| The significant features of the module include   * practical skill-oriented sessions on various aspects of the module * instructing in all three learning needs (visual, audio, and kinesthetics) to ensure diversity and inclusion * peer assessment and feedback * formative and summative assignments to measure student progress in the module |

|  |  |
| --- | --- |
| **Student profile in the context of this module:** | |
| What is the target group of students who would do this module? | Students who aspire to be career programmers, Software Engineers, Data Scientists, Web Developers, etc. |
| What **skills** should a *student* have **already** mastered before starting this Module? | Skills in C++ programming, organising, and problem-solving abilities. |
| What **prior knowledge** of the subject matter should a *student* have? | Design of algorithms, flow charts, and pseudo code. |

|  |  |
| --- | --- |
| **Non-expert support:** | |
| What **skills** and **prior knowledge** of the subject matter  should *tutors* have **already** mastered before starting to deliver this Module? | Advanced knowledge in computer programming, analysis of algorithms, and ability to apply concepts in the subject matter to solve real-world problems. |
| What **skills** do *support staff* need in order to support the delivery of this module? | Blended learning and e-moderating skills are important in addition to C++ programming skills. |

|  |  |  |
| --- | --- | --- |
| **Quality assurance matters** | | |
| How will feedback on module be obtained from students? | Feedback will be obtained by   * utilising tools such as Padlet to collect anonymous student feedback * end of semester student assessment of courses carried out by quality assurance and academic planning directorate * discussion forums on the LMS * assessment of student progress through formative and summative assignments | |
| How will student feedback be used to improve module? | Will lead to   * review and redesign of curriculum * enhance teaching, learning, and assessment methods * provision of learning resources | |
| A certificate, signed by the university’s Head of Quality Assurance, confirming that the module meets the requirements of the PEBL QA rubric is attached. | | Yes ❎ No ☐ |

UNIT LEVEL TEMPLATE

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit-level overview** | | **Unit** |  |
| Topic name: | **Review of C++ and Introduction to Data Structures and Algorithms** | | |
| Aim of the topic: | Review your knowledge of C++ programming language and introduce you to the Data Structures and Algorithms (DSA) module. | | |
| This topic covers: | A review of C++ programming language with emphasis on the aspects required for this module and introduce you to the Data Structures and Algorithms course. | | |
| Intended learning outcomes: | *At the end of this* ***topic****, you will be able to:*   1. describe the basic steps in C++ programming 2. explain the role of Data Structures and Algorithms (DSA) in software development. | | |

|  |  |
| --- | --- |
| Overview of student activity: | Read Chapter 1 of [Data Structures and Algorithms Made Easy](http://www.dhimangaurav.com/docs/data.pdf) and[Data Structures and Algorithms](http://www.dhimangaurav.com/docs/data.pdf) [in C++](http://160592857366.free.fr/joe/ebooks/ShareData/Data%20Structures%20and%20Algorithms%20in%20C++%202e%20By%20Michael%20Goodrich,%20Roberto%20Tamassia%20and%20David%20Mount.pdf). Watch the video of Chapter 1 of DSA made Easy and this [video](https://www.youtube.com/watch?v=z9bZufPHFLU) on DSA in C++. After reading and watching the videos   * Summarise the fundamental steps in C++ programming * Take notes of the fundamental concepts and essence of DSA   You will take an online quiz on C++ programming fundamentals in addition to a quiz on Introduction to Data Structures and Algorithms (E-tivity 3). |

|  |  |  |  |
| --- | --- | --- | --- |
| **Constructive alignment of unit level outcomes with module level outcomes, learning activities and assessment** *(Pressing <Tab> at the end of the table will provide additional rows in the table, if required.)* | | | |
| Intended unit learning outcomes: | No of module-level outcome | Activity where students engage with this outcome | Where and how is this outcome assessed? |
| ***At the end of this unit, you will be able to:*** | | | |
| 1. describe the basic steps in C++ programming | 1 | E-tivity 2 |  |
| 1. explain the role of Data Structures and Algorithms (DSA) in software development. | 1 | E-tivity 2, E-tivity 3 | Unit E-tivity 3 - An online quiz on C++ Programming and will be assessed online. |

|  |  |  |  |
| --- | --- | --- | --- |
| Detailed explanation of ALL student and teacher engagement with the unit:  ***(This should be presented in the order that the activities take place. So if students do work* online *before* *coming to the lecture, that should be shown ahead of what happens in class.***  ***If there is more than one opportunity for face-to-face contact, or more than one online task, there should be a separate section for each instance, and they should be presented in the template in the same order that students encounter them.)***  ***Content*** *– such as lecture material – can EITHER be shown here OR added as* ***clearly identifiable*** *addenda to the document. If you plan to use addenda, you should ensure that these are cross-referenced in this section.)* | | | |
| Module-level outcomes addressed: | | | |
| Module Level Outcome 1. | | | |
| Purpose of the unit: | | | |
| To ensure that you are up to date with the prerequisite of the module and receive a general overview and requirements of the DSA module. This unit activates prior knowledge. | | | |
| Over to you: *(a description of the process of the section)* | | | |
| In this Unit you will have a brief review of C++ programming. The review will highlight the aspects of the program that will help you to successfully complete this module. The unit will also introduce you to the concept of data structures and algorithms and their importance to you as a student as well the role they play in programming. The Data Structures and Algorithms (DSA) module will take you through solutions to standard problems in detail and give you an insight into how efficient it is to use each data structure. It will also teach you the science of evaluating the efficiency of an algorithm which will enable you to choose the best one for implementation when needed. The unit has three (3) E-tivities designed to help you achieve the aim of the unit. | | | |
| Pre-topic activity: | | Number of hours | 1 |
| Read Chapter 1: The Role of Algorithms in Computing (pages 5 - 15) of this [book](https://edutechlearners.com/download/Introduction_to_algorithms-3rd%20Edition.pdf). More explanation on the topic can be found from the link [Introduction to Data Structures and Algorithms](https://data-flair.training/blogs/introduction-to-data-structures-algorithms/). | | | |
| Face to face time: *(if applicable)* | | Number of hours | 2 |
| You will attend the first lecture of the course. The tutor will introduce him/herself to the class and ask you to introduce yourself as well. The tutor will give a couple of guidelines for successful completion of the course and will ask you to state any expectations you have of the module. You will also be introduced to the Learning Management System (LMS) where the online aspect of the module will be done. | | | |
| Online activity: | | Number of hours | 2 |
| What should students do? | **E-tivity 1**  Log in to the University e-learning platform (<https://elearning.uenr.edu.gh/login/index.php>) and navigate to the Data Structures and Algorithms course page.  Take the page tour to familiarise yourself with all the required aspects of the page. This will help you to kickstart your online activities.  **E-tivity 2**  Read Chapter 1 of [Data Structures and Algorithms Made Easy](http://www.dhimangaurav.com/docs/data.pdf), [Data Structures and Algorithms in C++](http://160592857366.free.fr/joe/ebooks/ShareData/Data%20Structures%20and%20Algorithms%20in%20C++%202e%20By%20Michael%20Goodrich,%20Roberto%20Tamassia%20and%20David%20Mount.pdf), and watch this [video](https://www.youtube.com/watch?v=z9bZufPHFLU). Then   * summarise the fundamental steps in C++ programming * take notes of the fundamental concepts and essence of DSA   **E-tivity 3**  Take this [online quiz](https://www.enggwave.com/c-pp-programming-test-1) on C++ Programming and Introduction to Data Structures and Algorithms (DSA). | | |
| Where do they do it? | **E-tivity 1**  The University E-Learning Platform.  **E-tivity 2**  You will use your laptop, PC, tablet, or phone to read and watch the video as well as write the summary and notes.  **E-tivity 3**  The University E-Learning Platform. | | |
| By when should they do it? | The e-moderator will give the specific data and time for completion of the e-tivities. All the e-tivities will be completed before next week’s class. | | |
| E-moderator/tutor role | | | |
| E-moderator will prepare the online quizzes, assist/support you (when needed) in accessing the e-learning platform, ensure the learning materials are available and accessible as well as assess E-tivity 3. | | | |
| How are the learning outcomes in this unit assessed? | | Number of hours | 1 |
| You will take a formative assessment quiz (Multiple Choice Questions) on C++ Programming and DSA concepts. | | | |
| How does this section link to other sections of the module? | | | |
| This section provides the foundation of the course required to comprehend the remaining units. | | | |

|  |  |
| --- | --- |
| = Total number of hours | 6 |

|  |  |
| --- | --- |
| **Some important questions** | |
| Which learning resources/ references will scaffold the students’ learning? | The following learning resources are provided for use in the entire module. [Data Structures and Algorithms Made Easy](http://www.dhimangaurav.com/docs/data.pdf), [Data Structures and Algorithms in C++](http://160592857366.free.fr/joe/ebooks/ShareData/Data%20Structures%20and%20Algorithms%20in%20C++%202e%20By%20Michael%20Goodrich,%20Roberto%20Tamassia%20and%20David%20Mount.pdf). The following are selected for this unit [Introduction to C++ in Data Structures](https://www.youtube.com/watch?v=z9bZufPHFLU),  [Introduction to Data Structures and Algorithms](https://data-flair.training/blogs/introduction-to-data-structures-algorithms/), and [Why Learn Data Structures and Algorithms](https://www.programiz.com/dsa/why-algorithms) |
| How are students enabled to access the resources? | Students are provided with internet service on campus and in university hostels. PDF versions of the learning resources are provided to students (for offline use) as well as online links to the resources and downloadable video resources. Students can also use the university e-library resources to assess the learning resources. |
| Where in this unit are students expected to work collaboratively? | N/A |
| How has an inclusive approach been incorporated in this unit? | Different versions of the learning resources are provided - PDF and video versions to cater to students with different learning needs. |
| How will feedback on unit be obtained from students? | Anonymously through Padlet, discussion forums, and assessment of the Chapter online quiz. |
| How will student feedback be used to improve unit? | Feedback will be forwarded to the course development team for consideration and implementation. The tutor will anonymously discuss and draw the students’ attention to key observations in class. |
| At which point(s) will students receive formative feedback on the work they have done in the unit? | You will receive feedback before the start of Week 2. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit-level overview** | | **Unit** |  |
| Topic name: | **Stacks** | | |
| Aim of the topic: | Introduce you to stacks and their implementation in C++. | | |
| This topic covers: | Both the theoretical and practical aspects of Stacks data structures. On the theoretical side, you will learn to identify and illustrate stacks. The practical aspect will cover stack implementation in C++. | | |
| Intended learning outcomes: | *At the end of this* ***topic****, you will be able to:*   1. identify and illustrate stacks 2. solve software requirements that require stacks | | |

|  |  |
| --- | --- |
| Overview of student activity: | Student subgroups will watch a YouTube video on [Stacks](https://www.youtube.com/watch?v=A3ZUpyrnCbM). Pay close attention and watch it as many times as necessary   * Give your original example of a circumstance in which you will use stacks. * Design an algorithm to demonstrate how the stack operations will function. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Constructive alignment of unit level outcomes with module level outcomes, learning activities and assessment** *(Pressing <Tab> at the end of the table will provide additional rows in the table, if required.)* | | | |
| Intended unit learning outcomes: | No of module-level outcome | Activity where students engage with this outcome | Where and how is this outcome assessed? |
| ***At the end of this unit, you will be able to:*** | | | |
| 1. identify and illustrate stacks | 1 | E-tivity 1 | Online peer assessment using the review guideline provided on the LMS |
| 1. solve software requirements that require stacks | 2 and 3 | E-tivity 2 | Online tutor assessment using the E-tivity Rubric on the LMS |

|  |  |  |  |
| --- | --- | --- | --- |
| Detailed explanation of ALL student and teacher engagement with the unit:  ***(This should be presented in the order that the activities take place. So if students do work* online *before* *coming to the lecture, that should be shown ahead of what happens in class.***  ***If there is more than one opportunity for face-to-face contact, or more than one online task, there should be a separate section for each instance, and they should be presented in the template in the same order that students encounter them.)***  ***Content*** *– such as lecture material – can EITHER be shown here OR added as* ***clearly identifiable*** *addenda to the document. If you plan to use addenda, you should ensure that these are cross-referenced in this section.)* | | | |
| Module-level outcomes addressed: | | | |
| Module level outcome 1, 2 and 3. | | | |
| Purpose of the unit: | | | |
| Explain the basic principle of stack data structures, perform push(), pop(), top(), size(), isfull() and empty() operations of stacks in C++ program and assess your programming skills and test the efficiency of your code. | | | |
| Over to you: *(a description of the process of the section)* | | | |
| In this unit you will learn all about Stacks data structures. The unit will cover introduction to stacks, stacks in C++ and Stack operations. In addition to attending a face-to-face lecture, you will do two e-tivities designed to help you identify problems or scenarios where stacks can be applied and write algorithms and pseudocode for the scenario. You will then write and compile a C++ code to solve the identified problem. Details are given in E-tivity 1 and 2 in the online activity section. | | | |
| Pre-topic activity: | | Number of hours |  |
| N/A | | | |
| Face to face time: *(if applicable)* | | Number of hours | 2 |
| Attend Week 3 lecture on Stacks. The lecture will cover the following topics   * Introduction to Stacks * Stacks in C++ * Stack operations | | | |
| Online activity: | | Number of hours | 3 |
| What should students do? | **E-tivity 1**  You will watch two videos:   1. [Introduction to stacks](https://youtu.be/r7P9sy5Rar8) 2. [Stack implementation C++](https://youtu.be/08QSylWv6jM)   You will then describe two scenarios (problems) where you think using stack data structure will be optimal for solving the problem. Write an algorithm that shows how the various operations of stacks will solve the problem. Post your descriptions and algorithm on the discussion board on the LMS on the date specified by the tutor.  You are to analyse/review the scenarios and algorithms posted by your peers and leave a comment/feedback and recommendation (if any) on how the algorithm could be optimised. Note that your E-tivity WILL NOT be marked complete if you don’t do this part.  You will follow these [Guidelines](https://drive.google.com/file/d/1An4izhrEHe0rX4vyM8wt3DB-qTaEbPof/view?usp=sharing) to review your peer’s work. These [peer review questions](https://drive.google.com/file/d/1Qtxcg7zlWDneqWsD4_XJ33UBd2lkUudy/view?usp=sharing) could also be of help.  **General guidelines for giving feedback**   * To begin, mention what you appreciate most about the work and inform your peer of areas in which he or she excelled. * Secondly, propose one or two ways in which you believe they could have done better. * You should employ a pleasant tone that is considerate of your peers' feelings. * If you're unsure, you can politely request clarification or comment nicely. For instance, I believe you may have... or I am of the view that...... * Present your comments by the specified deadlines to allow sufficient time for your colleagues to consider your suggestions if any. * Always remain focused on the assessment's objective.   **E-tivity 2**  Stacks implementation in C++  Based on the scenarios and algorithm written in E-tivity 1, write pseudocode and a C++ code to run the algorithm, taking into consideration recommendations by your peers.  Use this [online compiler](https://www.programiz.com/cpp-programming/online-compiler/) to run your C++ code.  Share your executed code in the LMS by taking screenshots of the outputs of the code and the execution time. Your code should compile without errors and execute within the shortest possible time. | | |
| Where do they do it? | You will do this during your study time on your PC or laptop and the LMS. | | |
| By when should they do it? | The timeline is specified by the tutor. But both e-tivities will be completed before the next class. | | |
| E-moderator/tutor role | | | |
| E-moderator will   * prepare the e-tivities * give you specific assistance/support (when needed) * ensure the learning materials are available and accessible * assess the e-tivities | | | |
| How are the learning outcomes in this unit assessed? | | Number of hours | 1 |
| E-tivity 2 Assessment will be based on this [Rubric](https://drive.google.com/file/d/1n72dAKkfq37OGMLfs5oKxUevC9AVsuAJ/view?usp=sharing). | | | |
| How does this section link to other sections of the module? | | | |
| The Unit links to Unit 3 because both Stacks and Queues are data structures that help store data in a way that is predefined according to the needs. The unit is also linked to Unit 6 where functions such as ‘call stacks’ are used in Recursion algorithms. | | | |

|  |  |
| --- | --- |
| = Total number of hours | 6 |

|  |  |
| --- | --- |
| **Some important questions** | |
| Which learning resources/ references will scaffold the students’ learning? | You will use both video and reading resources to scaffold their learning of Stacks data structures  The resources include   1. [Introduction to Stacks and Queues](https://www.youtube.com/watch?v=A3ZUpyrnCbM) 2. [Introduction to stacks](https://youtu.be/r7P9sy5Rar8) 3. [Stack implementation C++](https://youtu.be/08QSylWv6jM) 4. Stacks and Queues, Pages 232 - 236 of the book [Introduction to Algorithms](https://edutechlearners.com/download/Introduction_to_algorithms-3rd%20Edition.pdf) |
| How are students enabled to access the resources? | Students are provided with internet service on campus and in university hostels. You will be provided with links to the videos and readings. You can watch the videos and read the book online or you can download it for offline use. You could also print the section of Stacks and Queues of the reading resource. |
| Where in this unit are students expected to work collaboratively? | During the discussion forum when you will comment on at least the work of two of your mates. |
| How has an inclusive approach been incorporated in this unit? | Different versions of the learning resources are provided - PDF and video versions to cater to students with different learning needs. |
| How will feedback on unit be obtained from students? | Through your comments on your peer’s work on the discussion forum and Padlet. |
| How will student feedback be used to improve unit? | Feedback will be forwarded to the course development team for consideration and implementation. The tutor could also anonymously discuss and draw your attention to his/her key observations in class. |
| At which point(s) will students receive formative feedback on the work they have done in the unit? | You will receive feedback before the start of Week 3. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit-level overview** | | **Unit** |  |
| Topic name: | **Queues** | | |
| Aim of the topic: | Introduce you to queues data structure and explore its unique attributes and importance in programming. | | |
| This topic covers: | This unit will cover both the theoretical and practical aspects of Queues data structures. On the theoretical side you will learn the unique attributes and the importance of Queues in programming. The practical aspect will cover Queues implementation in C++. | | |
| Intended learning outcomes: | *At the end of this* ***topic****, you will be able to:*   1. identify and illustrate queues 2. solve software requirements problems that require the application of queues | | |

|  |  |
| --- | --- |
| Overview of student activity: | Student groups (each with members note more than 5) will watch a YouTube video on [Queues](https://www.youtube.com/watch?v=A3ZUpyrnCbM) and do the following   * give your own original example of a circumstance in which the use of queues is optimal * design an algorithm to demonstrate how the queues operations will function. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Constructive alignment of unit level outcomes with module level outcomes, learning activities and assessment** *(Pressing <Tab> at the end of the table will provide additional rows in the table, if required.)* | | | |
| Intended unit learning outcomes: | No of module-level outcome | Activity where students engage with this outcome | Where and how is this outcome assessed? |
| ***At the end of this unit, you will be able to:*** | | | |
| 1. identify and illustrate queues | 1 | E-tivity 1 | Assessment will be online using the E-tivity Rubric |
| 1. solve software requirements problems that require the application of queues | 2, 3 | E-tivity 2 | Assessment will be online using the E-tivity Rubric |

|  |  |  |  |
| --- | --- | --- | --- |
| Detailed explanation of ALL student and teacher engagement with the unit:  ***(This should be presented in the order that the activities take place. So if students do work* online *before* *coming to the lecture, that should be shown ahead of what happens in class.***  ***If there is more than one opportunity for face-to-face contact, or more than one online task, there should be a separate section for each instance, and they should be presented in the template in the same order that students encounter them.)***  ***Content*** *– such as lecture material – can EITHER be shown here OR added as* ***clearly identifiable*** *addenda to the document. If you plan to use addenda, you should ensure that these are cross-referenced in this section.)* | | | |
| Module-level outcomes addressed: | | | |
| Module 1 and 3 | | | |
| Purpose of the unit: | | | |
| This unit will teach you the different types of queue data structure, their basic operations, implementation, and queue applications in C++. | | | |
| Over to you: *(a description of the process of the section)* | | | |
| The unit is designed to give you an understanding of Queues data structures and their application in programming. A face-to-face lecture will give you an overview of Queue Data Structure, types of Queues in Data Structure, basic operations in Queue Data Structure, and Queue implementation. You will also have two e-tivities aimed at equipping you to achieve the learning objectives. | | | |
| Pre-topic activity: | | Number of hours | 1 |
| Read this text [Queue Data Structure: Types, Implementation, Applications](https://www.naukri.com/learning/articles/queue-data-structure-types-implementation-applications/) before attending the face-to-face class. | | | |
| Face to face time: *(if applicable)* | | Number of hours | 2 |
| Attend a two-hour face-to-face class on Queue data structures. The class will cover the following   * overview of Queue Data Structure * types of Queues in Data Structure * basic operations in Queue Data Structure * implementation of Queue * application of Queue in Data Structure | | | |
| Online activity: | | Number of hours | 2 |
| What should students do? | **E-tivity 1**  In this activity, each group will   1. watch this YouTube video on [Queues](https://www.youtube.com/watch?v=A3ZUpyrnCbM). Pay close attention and watch it as many times as necessary 2. give your original example of a circumstance in which the use of Queues is optimal 3. design an algorithm to demonstrate how the Queues operations will function in the identified circumstance.   **E-tivity 2**  In this E-tivity, you will   1. convert the algorithm you wrote in your groups to a C++ program and execute it on the [online compiler](https://www.programiz.com/cpp-programming/online-compiler/). 2. send your algorithm, C++ code, and the output of the compilation/execution to the tutor through email by a determined date.   **Note: As part of your submission please kindly add any personal and group observations you made during the e-tivities which will serve as feedback to improve the course.** | | |
| Where do they do it? | You will do this during your study time on your PC or laptop and the LMS. | | |
| By when should they do it? | The date will be communicated by the tutor. | | |
| E-moderator/tutor role | | | |
| E-moderator will   * prepare the e-tivities * give you specific assistance/support (when needed) * ensure the learning materials are available and accessible * assess the e-tivities | | | |
| How are the learning outcomes in this unit assessed? | | Number of hours | 1 |
| Assessment will be based on the [Rubric](https://drive.google.com/file/d/1AhtLYPwH3-dnDxQXnLtA0JxFClEX6-gT/view?usp=sharing) | | | |
| How does this section link to other sections of the module? | | | |
| The Unit links to Unit 2 because both Stacks and Queues are data structures that help store data in a way that is predefined according to the needs. The Unit also links to Unit 4 because a Queue data structure can be implemented using a one-dimensional array. | | | |

|  |  |
| --- | --- |
| = Total number of hours | 6 |

|  |  |
| --- | --- |
| **Some important questions** | |
| Which learning resources/ references will scaffold the students’ learning? | You will use both video and reading resources to scaffold your learning of Queues.  The resources include   1. [Queue Data Structure: Types, Implementation, Applications](https://www.naukri.com/learning/articles/queue-data-structure-types-implementation-applications/) 2. YouTube video on [Queues](https://www.youtube.com/watch?v=A3ZUpyrnCbM) 3. Stacks and Queues, Pages 232 - 236 of the book [Introduction to Algorithms](https://edutechlearners.com/download/Introduction_to_algorithms-3rd%20Edition.pdf) |
| How are students enabled to access the resources? | Students are provided with internet service on campus and in university hostels. You are provided with links to the videos and readings. They can watch the videos and read the book online or they can download it for offline use. Students can print the section of Queues of the reading resource. |
| Where in this unit are students expected to work collaboratively? | Both e-tivities for the week are collaborative/group work. Student groups will collaborate to work on the e-tivities and submit a response per group. |
| How has an inclusive approach been incorporated in this unit? | Different versions of the learning resources are provided - PDF and video versions to cater to students with different learning needs. |
| How will feedback on unit be obtained from students? | As part of their group work submission, students are encouraged to add a note of any observations/concerns they have about the e-tivities. This will serve as feedback for the unit. |
| How will student feedback be used to improve unit? | Feedback will be forwarded to the course development team for consideration and implementation where appropriate. |
| At which point(s) will students receive formative feedback on the work they have done in the unit? | Students will receive feedback before the start of Week 4. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit-level overview** | | **Unit** |  |
| Topic name: | **Arrays and Linked Lists** | | |
| Aim of the topic: | For you to understand and differentiate between the principles and operations of linked lists and be able to apply arrays and linked lists in solving problems in programming. | | |
| This topic covers: | This unit is designed to cover both the theoretical and practical aspects of arrays and linked lists data structures. Theoretically, you will learn to differentiate between arrays and the principles and operations of the different types of linked lists. The practical aspect will involve the application of arrays and linked lists in solving a real-life problem. | | |
| Intended learning outcomes: | *At the end of this* ***topic****, you will be able to:*   1. differentiate between the principles and operation of the various linked lists 2. apply arrays and linked lists in solving problems | | |

|  |  |
| --- | --- |
| Overview of student activity: | Read Chapter 3 of [DSA made Easy](http://www.dhimangaurav.com/docs/data.pdf), Chapter 3, sections 1 to 4 [DSA in C++](http://160592857366.free.fr/joe/ebooks/ShareData/Data%20Structures%20and%20Algorithms%20in%20C++%202e%20By%20Michael%20Goodrich,%20Roberto%20Tamassia%20and%20David%20Mount.pdf), and watch this video titled [Data Structures: Arrays vs Linked Lists](https://www.youtube.com/watch?v=lC-yYCOnN8Q)   * Diagrammatically describe the fundamental operations on arrays and various linked lists * Replay the video to be certain you did not miss anything |

|  |  |  |  |
| --- | --- | --- | --- |
| **Constructive alignment of unit level outcomes with module level outcomes, learning activities and assessment** *(Pressing <Tab> at the end of the table will provide additional rows in the table, if required.)* | | | |
| Intended unit learning outcomes: | No of module-level outcome | Activity where students engage with this outcome | Where and how is this outcome assessed? |
| ***At the end of this unit, you will be able to:*** | | | |
| 1. differentiate between the principles and operation of the various linked lists | 1 | Face-to-face  lecture | The tutor will assess and grade your input accordingly |
| 1. apply arrays and linked lists in solving problems | 1 | E-tivity 2 | The tutor will use the rubric to award marks on completion of the E-tivity online using the LMS |

|  |  |  |  |
| --- | --- | --- | --- |
| Detailed explanation of ALL student and teacher engagement with the unit:  ***(This should be presented in the order that the activities take place. So if students do work* online *before* *coming to the lecture, that should be shown ahead of what happens in class.***  ***If there is more than one opportunity for face-to-face contact, or more than one online task, there should be a separate section for each instance, and they should be presented in the template in the same order that students encounter them.)***  ***Content*** *– such as lecture material – can EITHER be shown here OR added as* ***clearly identifiable*** *addenda to the document. If you plan to use addenda, you should ensure that these are cross-referenced in this section.)* | | | |
| Module-level outcomes addressed: | | | |
| Module Level Outcomes 1, 2 and 3 | | | |
| Purpose of the unit: | | | |
| To ensure that students are up to date with the prerequisite of the module and receive a general overview and requirements of the DSA module. This unit activates prior knowledge. | | | |
| Over to you: *(a description of the process of the section)* | | | |
| In the unit, you will learn about arrays and linked lists, and the similarities and differences between them. You will also learn what and how operations can be performed on various linked lists. In addition, you will learn how to apply arrays and linked lists in solving real-world problems. There will be face-to-face as well as e-tivities. You will review the response to E-tivity 1 of three of your peers and use the feedback guidelines to provide the appropriate feedback on their work. E-tivity 2 will involve identifying a scenario that will require the application of arrays and linked lists. The tutor will grade your response based on the activity rubric. Do your best to complete and submit your e-tivities before the deadline you will be given. | | | |
| Pre-topic activity: | | Number of hours | 1 |
| Read Chapter 3 of [DSA made Easy](http://www.dhimangaurav.com/docs/data.pdf), Chapter 3, sections 1 to 4 [DSA in C++](http://160592857366.free.fr/joe/ebooks/ShareData/Data%20Structures%20and%20Algorithms%20in%20C++%202e%20By%20Michael%20Goodrich,%20Roberto%20Tamassia%20and%20David%20Mount.pdf) in groups, students should identify and discuss real-world scenarios where arrays and linked lists can be applied | | | |
| Face to face time: *(if applicable)* | | Number of hours | 2 |
| Attend Week 2 lecture on Arrays and Linked Lists. This class session will cover the following   * introduction to linked lists and arrays * linked list vs arrays * linked list insertion * linked list deletion * one dimensional array * two-dimensional arrays * multidimensional arrays * operations on arrays and linked lists, etc.   You will learn how to diagrammatically describe the fundamental operations on arrays and various linked lists. | | | |
| Online activity: | | Number of hours | 2 |
| What should students do? | **E-tivity 1**   * Watch this video titled [Data Structures: Arrays vs Linked Lists](https://www.youtube.com/watch?v=lC-yYCOnN8Q). * Create diagrams of arrays and linked lists and paste them on the Chapter page on Moodle * Review and give feedback to at least three of your peers on their diagrams. Your E-tivity will not be marked complete if this review and feedback are not done   **Use the guideline below to review your Peer’s work**  **Student guidelines for peer review**   * Make sure to read the entire assignment – code, documentation, and analysis all the way through before you make any comments. * Keep in mind the assignment deadlines. Do the assessment early, a day or two before the deadline to allow time for your peer to respond to your comment as well as edit his/her assignment before the deadline. * Point out the strengths as well as the weaknesses of the assignment. * Offer suggestions, not commands. * Comments should be appropriate and constructive. There is no need to be rude. Be respectful and considerate of your peer’s feelings. * Ensure that your comments are clear and text-specific so that your peer will know what you are referring to. * When necessary, raise questions that cross your mind, points that may have not occurred to your peer author. * Try not to overwhelm your peers with too much commentary. * Be careful not to let your own opinions bias your review (for example, do not suggest that your peer completely rewrite the paper just because you do not agree with his/her point of view). * Reread your comments for revision before passing them on to your peer. Make sure all your comments make sense, are relevant to the assignment, useful and are easy to follow. * Avoid turning your peer's paper into your paper.   **General guidelines for giving feedback**   * To begin, mention what you appreciate most about the work and inform your peer of areas in which he or she excelled. * Secondly, propose one or two ways in which you believe they could have done better. * You should employ a pleasant tone that is considerate of your peers' feelings. * If you are unsure, you can politely request clarification or comment nicely. For instance, I believe you may have... or I am of the view that...... * Present your comments by the specified deadlines to allow sufficient time for your colleagues to consider your suggestions if any. * Always remain focused on the assessment's objective.   **Note: While peer grades do not affect your final grade, they are crucial in determining your progress towards attaining the course's learning outcomes. Relax and take pleasure in the feedback.**  **E-tivity 2**  The following group of friends took part in a video game and attained the following scores:   * Anna - Game score: 1134 * Kojo - Game score: 600 * Ama - Game Score: 400 * John – Game score 1000  1. Write a C++ program using arrays to store the game scores. Now insert Paul’s game score of 200 between Anna and Kojo and then delete John’s Score. Use [this online compiler](https://www.programiz.com/cpp-programming/online-compiler/) for your work. 2. Using the same group of people, use **singly**, **doubly** and **circular** lists to store the information. And perform some basic operations on them.   Copy your code and output of the compilation and paste on the assignment section in the Unit assignment section on the LMS. | | |
| Where do they do it? | You will use your laptop or personal computer for the e-tivities | | |
| By when should they do it? | All the e-tivities will be completed by an agreed date, usually before the next week’s class. | | |
| E-moderator/tutor role | | | |
| E-moderator will   * prepare the e-tivities * give you specific assistance/support (when needed) * ensure the learning materials are available and accessible * assess the e-tivities | | | |
| How are the learning outcomes in this unit assessed? | | Number of hours | 1 |
| Learning outcomes will be assessed using the E-tivity 2 [Rubric](https://drive.google.com/file/d/12DDpddRloHrT9_NpJLZLjzrvFSWUBFJU/view?usp=sharing). | | | |
| How does this section link to other sections of the module? | | | |
| The Unit links to Unit 3 and 2 because Queue and Stacks data structures can be implemented with arrays. | | | |

|  |  |
| --- | --- |
| = Total number of hours | 6 |

|  |  |
| --- | --- |
| **Some important questions** | |
| Which learning resources/ references will scaffold the students’ learning? | This [website](https://www.cs.usfca.edu/~galles/visualization/QueueArray.html) will aid you to simulate the operations of arrays and this video on [Data Structures: Arrays vs Linked Lists](https://www.youtube.com/watch?v=lC-yYCOnN8Q) are the learning resources for this unit. |
| How are students enabled to access the resources? | Students are provided with internet service on campus and in university hostels. Resources for this unit are available in both online and offline modes. PDFs can be downloaded. Students can use any smart device to access resources. |
| Where in this unit are students expected to work collaboratively? | The collaborative work is done before class in groups after they have individually completed the reading assignment. |
| How has an inclusive approach been incorporated in this unit? | Students have been provided with different types of reading materials, text, video, and graphical representations of concepts. |
| How will feedback on unit be obtained from students? | Student comments on the peer review exercise will provide insights into the understanding of arrays and linked lists. Constructive comments will indicate some level of understanding. |
| How will student feedback be used to improve unit? | Feedback will be used to improve sections of the module where appropriate. |
| At which point(s) will students receive formative feedback on the work they have done in the unit? | You will receive feedback on the diagrams you posted online and on the exercise on short code interpretation before the start of Week 5. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit-level overview** | | **Unit** |  |
| Topic name: | **Hashing technique and Linear probing** | | |
| Aim of the topic: | Introduce you to hashing techniques and learning probing, the basic operations of these data structures, and their significance in programming. | | |
| This topic covers: | This unit will cover both the theoretical and practical aspects of hashing technique and Linear probing data structures. On the theoretical side, you will learn the unique attributes, how to perform basic operations with hashing technique and linear probing, and their importance in programming. The practical aspect will include the implementation of hashing technique and linear probing in C++ | | |
| Intended learning outcomes: | *At the end of this* ***topic****, you will be able to:*   1. describe hashing technique and linear probing 2. implement hashing technique in a programming task 3. apply linear probing to solve collision problems in hashing | | |

|  |  |
| --- | --- |
| Overview of student activity: | You will do the following activities designed to achieve the unit’s learning outcomes. They include   1. a pre-topic activity 2. identifying a real-world problem that requires the application of hashing and 3. writing a C++ program to solve the identified problem |

|  |  |  |  |
| --- | --- | --- | --- |
| **Constructive alignment of unit level outcomes with module level outcomes, learning activities and assessment** *(Pressing <Tab> at the end of the table will provide additional rows in the table, if required.)* | | | |
| Intended unit learning outcomes: | No of module-level outcome | Activity where students engage with this outcome | Where and how is this outcome assessed? |
| ***At the end of this unit, you will be able to:*** | | | |
| 1. describe hashing technique and linear probing | 1 | Pre-topic activity | Will not be assessed |
| 1. implement hashing technique in a programming task | 2 | E-tivity 1 | Assessment will be online using E-tivity 1 Rubric |
| 1. apply linear probing to solve collision problems in hashing | 3 | E-tivity 2 | Assessment will be online using E-tivity 2 Rubric |

|  |  |  |  |
| --- | --- | --- | --- |
| Detailed explanation of ALL student and teacher engagement with the unit:  ***(This should be presented in the order that the activities take place. So if students do work* online *before* *coming to the lecture, that should be shown ahead of what happens in class.***  ***If there is more than one opportunity for face-to-face contact, or more than one online task, there should be a separate section for each instance, and they should be presented in the template in the same order that students encounter them.)***  ***Content*** *– such as lecture material – can EITHER be shown here OR added as* ***clearly identifiable*** *addenda to the document. If you plan to use addenda, you should ensure that these are cross-referenced in this section.)* | | | |
| Module-level outcomes addressed: | | | |
| Module Outcomes 1, 2, and 3. | | | |
| Purpose of the unit: | | | |
| The unit is designed to impart the knowledge and expertise of hashing techniques and linear probing in programming and their application programming. | | | |
| Over to you: *(a description of the process of the section)* | | | |
| In this unit, you have both face-to-face lectures and a couple of e-tivities. You will first do a pre-topic activity by watching a YouTube video. After watching the video, you will write down short notes on what you learned and present them during the face-to-face lecture. Later will identify a real-world problem and explain how hashing will apply to the problem and subsequently write a C++ code to solve the problem identified with hashing. Details are provided in the Pre-topic activity, E-tivity 1, and E-tivity 2. | | | |
| Pre-topic activity: | | Number of hours | 1 |
| You will watch this [video](https://www.youtube.com/watch?v=wWgIAphfn2U&t=30s) on Hashing and Linear probing and prepare not more than a half-page notes on what you will learn. Bring your prepared note to the face-to-face class. | | | |
| Face to face time: *(if applicable)* | | Number of hours | 2 |
| In a 2-hour face-to-face lecture, the tutor will ask some of you to read to the class your output from the pre-topic activity. Then there will be a lecture to cover the following   * introduction to hashing and linear probing * application of hashing and linear probing * primary operations on a hash table - search, insert, delete | | | |
| Online activity: | | Number of hours | 2 |
| What should students do? | **E-tivity 1**   1. Hashing is vital when it comes to security and data privacy in computing. However, there are three primary requirements in implementing a good hash function for a given data type. Outline these requirements and explain their importance in hash functions. 2. Provide a simple example of a Hash Function   **E-tivity 2**   1. Identify a real-world problem that will require the application of hashing. Explain why you think hashing will be effectively applied to the identified problem. Send your response, as a PDF file, to the tutor’s provided email on the deadline that he/she will give. 2. Write a C++ program to solve the problem identified in E-tivity 1. Use the [online compiler](https://www.programiz.com/cpp-programming/online-compiler/) to run the code. Submit a screenshot of the result, and execution time, and write code to the tutor’s email before the given deadline. | | |
| Where do they do it? | You will do these e-tivities on your own using your laptop or personal computer. | | |
| By when should they do it? | The tutor/e-moderator will provide the submission deadlines. | | |
| E-moderator/tutor role | | | |
| N/A | | | |
| How are the learning outcomes in this unit assessed? | | Number of hours | 1 |
| E-tivity 1 will be assessed with this [Rubric](https://drive.google.com/file/d/1GjJCbvCGDa2Qtua0WnmmKbEIh6Egln-n/view?usp=sharing)  E-tivity 2 will be assessed with this [Rubric](https://drive.google.com/file/d/1I4jDsJ4eGwdQisrtkW3QZEiQFG1Zrppn/view?usp=sharing) | | | |
| How does this section link to other sections of the module? | | | |
| In a hash table, data is stored in an array format, where each data value has its unique index value hence this unit has a direct link to Unit 4 as well as Units 2 and 3. | | | |

|  |  |
| --- | --- |
| = Total number of hours | 6 |

|  |  |
| --- | --- |
| **Some important questions** | |
| Which learning resources/ references will scaffold the students’ learning? | You will utilise both video and reading resources to scaffold your learning of Queues.  The resources include   1. [Hashing](https://www.youtube.com/watch?v=wWgIAphfn2U&t=30s) 2. [Linear Probing](http://web.stanford.edu/class/archive/cs/cs166/cs166.1166/lectures/12/Small12.pdf) 3. Hash Tables, Pages 253 - 285 of the book [Introduction to Algorithms](https://edutechlearners.com/download/Introduction_to_algorithms-3rd%20Edition.pdf) |
| How are students enabled to access the resources? | They are provided with links to the videos and readings. They can watch the videos and read the book online or they can download it for offline use. Students can print the section of Hashing and Linear Probing of the reading resource. |
| Where in this unit are students expected to work collaboratively? | N/A |
| How has an inclusive approach been incorporated in this unit? | Different versions of the learning resources are provided - PDF and video versions to cater to students with different learning needs. |
| How will feedback on unit be obtained from students? | As part of their unit’s E-tivity results submission, students are encouraged to add a note of any observations/concerns they have about the e-tivities. Together with student performance in the e-tivities, they will serve as feedback for the unit. |
| How will student feedback be used to improve unit? | Feedback will be forwarded to the course development team for consideration and implementation where appropriate. |
| At which point(s) will students receive formative feedback on the work they have done in the unit? | Students will receive feedback before the start of Week 6. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit-level overview** | | **Unit** |  |
| Topic name: | **Recursion** | | |
| Aim of the topic: | Introduce you to recursion data structures and illustrate the different types of recursions and their importance in programming. | | |
| This topic covers: | Both the theoretical and practical aspects of Recursion. On the theoretical side, you will learn the unique attributes and the importance of recursion data structures in programming while the practical aspect involves the application of the various kinds of recursion in solving software problems. | | |
| Intended learning outcomes: | *At the end of this* ***topic****, you will be able to:*   1. understand, explain and describe recursion and its relevance 2. how to develop recursive algorithms and programs | | |

|  |  |
| --- | --- |
| Overview of student activity: | You will be engaged in the following activities designed to achieve the unit’s learning outcomes. They include   1. attending a face-to-face lecture 2. performing E-tivity 1 and 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Constructive alignment of unit level outcomes with module level outcomes, learning activities and assessment** *(Pressing <Tab> at the end of the table will provide additional rows in the table, if required.)* | | | |
| Intended unit learning outcomes: | No of module-level outcome | Activity where students engage with this outcome | Where and how is this outcome assessed? |
| ***At the end of this unit, you will be able to:*** | | | |
| 1. understand, explain and describe recursion and its relevance | 1 | Pre-topic Activity and Face-to-face lecture, E-tivity 1 | Online with E-tivity 1 answer key |
| 1. how to develop recursive algorithms and programs | 2 | E-tivity 2 | Online on the LMS using the activity Rubric |

|  |  |  |  |
| --- | --- | --- | --- |
| Detailed explanation of ALL student and teacher engagement with the unit:  ***(This should be presented in the order that the activities take place. So if students do work* online *before* *coming to the lecture, that should be shown ahead of what happens in class.***  ***If there is more than one opportunity for face-to-face contact, or more than one online task, there should be a separate section for each instance, and they should be presented in the template in the same order that students encounter them.)***  ***Content*** *– such as lecture material – can EITHER be shown here OR added as* ***clearly identifiable*** *addenda to the document. If you plan to use addenda, you should ensure that these are cross-referenced in this section.)* | | | |
| Module-level outcomes addressed: | | | |
| Module level outcome 1 and 2. | | | |
| Purpose of the unit: | | | |
| The unit is designed to help you get the knowledge and skill in analysing and applying different types of recursion in algorithm design and implementation in C++. | | | |
| Over to you: *(a description of the process of the section)* | | | |
| In this unit, you have both face-to-face lectures and a couple of e-tivities. You will first do a pre-topic activity by watching a video on Recursion. You then attend a 2-hour face-to-face lecture. The tutor will ask you to give a summary of what you learned in the video tutorial before the lecture starts. A quiz, to be completed in E-tivity 1 has been designed to assess your understanding and appreciation of the concept of Recursion. E-tivity 2 will allow you to put your knowledge of Recursion into practice by writing an algorithm and a program in C++ that utilises recursion. | | | |
| Pre-topic activity: | | Number of hours | 1 |
| Open this [link](https://www.simplilearn.com/tutorials/cpp-tutorial/what-is-recursion-in-cpp?source=sl_frs_nav_playlist_video_clicked) to watch the video on Recursion in C++, its working, and examples. | | | |
| Face to face time: *(if applicable)* | | Number of hours | 2 |
| You will attend a face-to-face lecture where you will learn   1. basics of recursion 2. recursion properties 3. recursion implementation 4. analysis of recursion 5. types of recursion | | | |
| Online activity: | | Number of hours | 2 |
| What should students do? | **E-tivity 1**  Individual assignment   1. A word is considered ***elfish*** if it contains the letters: ***e***, ***l***, and ***f*** in it, in any order. So, for example, we would say that the following words are elfish: whiteleaf, tasteful, unfriendly, and waffles, because they each contain those letters.   i. Write a predicate function in C++ called ***elfish*** that, given a word, tells us if that word is elfish or not.  ii. Write a more generalised predicate function in C++ called ***x-ish*** that, given two words, returns true if all the letters of the first word are contained in the second.  **E-tivity 2**  Kofi and Ama are engaged in a game involving a collection of coins. The participants take turns selecting several coins from the group. Each time, a player may select one, two, or four coins, and the one who obtains the final coin is the winner. Assume that both players are extremely intelligent and will make every effort to devise a plan to win the game. For instance, if there are two coins and Kofi is the first player to choose, she will certainly choose two coins and win. If there are three coins and Kofi is still the first player to select, regardless of whether she picks one or two coins, Ama will win the game by picking the last coin. Given the number of coins and the sequence in which players select the coins (which implies the first and second players pick the coins), you are expected to create a software that will compute the winner of the game and the number of strategies for him/her to win. You should address the problem through recursion, with the parameters being read from the command line. Assume there are no more than thirty coins.  Submit your answer to the tutor by the deadline provided on the LMS. | | |
| Where do they do it? | On the LMS with their laptops and personal computers. | | |
| By when should they do it? | By the deadline, the tutor will give. | | |
| E-moderator/tutor role | | | |
| E-moderator will   * prepare the e-tivities * give you specific assistance/support (when needed) * ensure the learning materials are available and accessible * assess the e-tivities | | | |
| How are the learning outcomes in this unit assessed? | | Number of hours | 1 |
| The tutor will prepare an answer key to assess E-tivity 1.  The tutor can use the [E-tivity 2 Rubric](https://drive.google.com/file/d/12A-p52wUU-Dg_X19GYrwCKj2W0xG5JCZ/view?usp=sharing) or create one Rubric for assessment. | | | |
| How does this section link to other sections of the module? | | | |
| Recursive techniques can be utilised in sorting and searching algorithms; hence this Unit is directly linked to Units 8 and 9. This Unit is also linked to Unit 2 where functions link ‘call stack’ are utilised in Recursion. | | | |

|  |  |
| --- | --- |
| = Total number of hours | 6 |

|  |  |
| --- | --- |
| **Some important questions** | |
| Which learning resources/ references will scaffold the students’ learning? | The following resources on recursion have been selected to aid your learning of the unit.  The resources include   1. [C++ Recursion](https://www.simplilearn.com/tutorials/cpp-tutorial/what-is-recursion-in-cpp?source=sl_frs_nav_playlist_video_clicked) 2. [Recursion in C++ with examples and code](https://favtutor.com/blogs/recursion-cpp) 3. Recursion on Pages 141 - 152 of the book [Data Structures and Algorithms in C++ 2e](http://160592857366.free.fr/joe/ebooks/ShareData/Data%20Structures%20and%20Algorithms%20in%20C++%202e%20By%20Michael%20Goodrich,%20Roberto%20Tamassia%20and%20David%20Mount.pdf) 4. [Understanding Recursion Using Real-World Examples](https://www.byte-by-byte.com/understanding-recursion/) |
| How are students enabled to access the resources? | Students are provided with internet service on campus and in university hostels. They are provided with links to the videos and readings. They can watch the videos and read the book online or they can download it for offline use. Students can print the section of Recursion of the reading resource. |
| Where in this unit are students expected to work collaboratively? | N/A |
| How has an inclusive approach been incorporated in this unit? | Different versions of the learning resources are provided - PDF and video versions to cater to students with different learning needs. |
| How will feedback on unit be obtained from students? | As part of their unit’s E-tivity results submission, students are encouraged to add a note of any observations/concerns they have about the e-tivities. Together with student performance in the e-tivities, they will serve as feedback for the unit. |
| How will student feedback be used to improve unit? | Feedback will be forwarded to the course development team for consideration and implementation where appropriate. |
| At which point(s) will students receive formative feedback on the work they have done in the unit? | Students will receive feedback before the start of Week 7. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit-level overview** | | **Unit** |  |
| Topic name: | **Mid Semester Examinations** | | |
| Aim of the topic: | To test the understanding, recall, analysis, and application of lessons in Units 1 - 6 of the module. | | |
| This topic covers: | A formative assessment test on major aspects of the Units 1 -6 of the module. | | |
| Intended learning outcomes: | *At the end of this* ***topic****, you will be able to:*   1. personally assess how far or close you are from achieving the module learning outcomes 2. assess if you were able to achieve the learning outcomes of previous units | | |

|  |  |
| --- | --- |
| Overview of student activity: | You will write a one-hour formative assessment exam covering major aspects of Units 1 - 6 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Constructive alignment of unit level outcomes with module level outcomes, learning activities and assessment** *(Pressing <Tab> at the end of the table will provide additional rows in the table, if required.)* | | | |
| Intended unit learning outcomes: | No of module-level outcome | Activity where students engage with this outcome | Where and how is this outcome assessed? |
| ***At the end of this unit, you will be able to:*** | | | |
| 1. assess how far or close you are from achieving the module learning outcomes | 1, 2, 3 | Mid-semester examination | Offline formative mid-semester assessment |
| 1. assess if you were able to achieve the learning outcomes of previous units | 1, 2, 3 | Mid-semester examination | Offline formative mid-semester assessment |

|  |  |  |  |
| --- | --- | --- | --- |
| Detailed explanation of ALL student and teacher engagement with the unit:  ***(This should be presented in the order that the activities take place. So if students do work* online *before* *coming to the lecture, that should be shown ahead of what happens in class.***  ***If there is more than one opportunity for face-to-face contact, or more than one online task, there should be a separate section for each instance, and they should be presented in the template in the same order that students encounter them.)***  ***Content*** *– such as lecture material – can EITHER be shown here OR added as* ***clearly identifiable*** *addenda to the document. If you plan to use addenda, you should ensure that these are cross-referenced in this section.)* | | | |
| Module-level outcomes addressed: | | | |
| Module-level outcomes 1, 2, 3 | | | |
| Purpose of the unit: | | | |
| Assess students on their performance and ability to achieve the module and unit-level learning outcomes, allow tutors to assess and adjust learning outcomes and activities if necessary, and plan to achieve the module-level learning outcomes. | | | |
| Over to you: *(a description of the process of the section)* | | | |
| Revise all the units you have covered in class as well as all the activities in which you have engaged. Get a grasp of all the major aspects of the unit lessons to be in good standing for the assessment. Arrive at the examination hall on time and adhere to all examination rules and regulations during the exams. Good luck. | | | |
| Pre-topic activity: | | Number of hours | 4 |
| Revision of all activities and notes of Unit 1 - 6. | | | |
| Face to face time: *(if applicable)* | | Number of hours | 1 |
| A one-hour formative assessment test. | | | |
| Online activity: | | Number of hours |  |
| What should students do? | N/A | | |
| Where do they do it? | N/A | | |
| By when should they do it? | N/A | | |
| E-moderator/tutor role | | | |
| Assist you with revision if needed and help you prepare for the mid-semester examination. | | | |
| How are the learning outcomes in this unit assessed? | | Number of hours | 1 |
| The tutor will design an appropriate marking scheme to grade the examination.  However, pay attention to the following   * demonstrate clarity of thought and expression in your writing * produce scripts with no to minimal spelling and grammar errors * organise your work properly with appropriate numbering and indentations * avoid too many cancellations | | | |
| How does this section link to other sections of the module? | | | |
| This unit links directly to Units 1-6 and prepares you for the rest of the units in the module. | | | |

|  |  |
| --- | --- |
| = Total number of hours | 6 |

|  |  |
| --- | --- |
| **Some important questions** | |
| Which learning resources/ references will scaffold the students’ learning? | You will use all the reading texts and video resources used for Units 1 - 6 to you in preparation for the mid-semester exams. |
| How are students enabled to access the resources? | All reading and video learning resources of Units 1 to 6 have been made available to you. |
| Where in this unit are students expected to work collaboratively? | N/A |
| How has an inclusive approach been incorporated in this unit? | Provision will be made for students with any form of disability to write the examination in an inclusive environment. |
| How will feedback on unit be obtained from students? | Your performance in the mid-semester exam will be the main source of feedback for this unit. You are also allowed to give information on any personal observations you made during the exam to the tutor. |
| How will student feedback be used to improve unit? | Results and personal feedback from students will be analysed. Observed trends and insights will be a source of feedback for implementation where applicable. |
| At which point(s) will students receive formative feedback on the work they have done in the unit? | As soon as practicable. Preferably, two weeks after the mid-semester examinations. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit-level overview** | | **Unit** |  |
| Topic name: | **Sorting Algorithms** | | |
| Aim of the topic: | For you to learn about sorting algorithms and their various types and real applications as well as their implementation in the C++ programming language. | | |
| This topic covers: | The theory, practice, and application of various sorting algorithms. The theory will introduce you to the concept and principle of sorting and the differences between the various types of sorting algorithms. The practice and implementation will delve into writing sorting algorithms and implementing them in C++ as well as identifying and analysing situations where sorting will be applicable. | | |
| Intended learning outcomes: | *At the end of this* ***topic****, you will be able to:*   1. understand the principle and concept of sorting 2. write and code sorting algorithms in C++ 3. differentiate between the various sorting algorithms 4. real-life application of sorting algorithms | | |

|  |  |
| --- | --- |
| Overview of student activity: | In this Unit, you will be engaged in the following activities designed to achieve the unit’s learning outcomes. They include   1. attending a face-to-face lecture 2. performing E-tivity 1, 2 and 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Constructive alignment of unit level outcomes with module level outcomes, learning activities and assessment** *(Pressing <Tab> at the end of the table will provide additional rows in the table, if required.)* | | | |
| Intended unit learning outcomes: | No of module-level outcome | Activity where students engage with this outcome | Where and how is this outcome assessed? |
| ***At the end of this unit, you will be able to:*** | | | |
| 1. Understand the principle and concept of sorting | 1 | E-tivity 1 | Online using the E-tivity Rubric |
| 1. Write and code sorting algorithms in C++ | 1 | E-tivity 1 | Online using the E-tivity Rubric |
| 1. Differentiate between the various sorting algorithms | 1 | E-tivity 1 | Online using the E-tivity Rubric |
| 1. Real-life application of sorting algorithms | 2 and 3 | E-tivity 3 | Online using the E-tivity Rubric |

|  |  |  |  |
| --- | --- | --- | --- |
| Detailed explanation of ALL student and teacher engagement with the unit:  ***(This should be presented in the order that the activities take place. So if students do work* online *before* *coming to the lecture, that should be shown ahead of what happens in class.***  ***If there is more than one opportunity for face-to-face contact, or more than one online task, there should be a separate section for each instance, and they should be presented in the template in the same order that students encounter them.)***  ***Content*** *– such as lecture material – can EITHER be shown here OR added as* ***clearly identifiable*** *addenda to the document. If you plan to use addenda, you should ensure that these are cross-referenced in this section.)* | | | |
| Module-level outcomes addressed: | | | |
| Module-level outcomes 1, 2, and 3. | | | |
| Purpose of the unit: | | | |
| Sorting algorithms play a vital role in Data Structures and Algorithms. The knowledge that you will gain from this unit will help improve your thinking process about programming and software design as well as your coding style. The practice and application examples of this unit will cover a subset of situations that a software engineer is likely to encounter across the course of a career. The unit will also help you to know when and how these algorithms work well and when they do not, to ensure their proper usage. | | | |
| Over to you: *(a description of the process of the section)* | | | |
| This unit is designed to give you the knowledge and practical experience of applications of sorting algorithms that will enhance your skills and knowledge as a future software engineer. You will learn selection sort, bubble sort, insertion sort, merge sort, and their order of complexities and implementation in C++. The concepts can however apply to any programming language. You are provided learning resources that will enrich your learning experience. The face-to-face and online activity sections detail what you will be doing during the unit. | | | |
| Pre-topic activity: | | Number of hours |  |
| N/A | | | |
| Face to face time: *(if applicable)* | | Number of hours | 2 |
| You will attend a 2-hour face-to-face lecture. This lecture will cover the following   * Simple Sorting Algorithm * Selection Sort and its diagrammatic explanation * Bubble Sort and its graphical explanation * Insertion Sort and graphical implementation * An extensive explanation of Merge Sort * Differences between the Algorithms | | | |
| Online activity: | | Number of hours | 3 |
| What should students do? | **E-tivity 1**  In this E-tivity your tutor will give you a quiz that will test your understanding of the concept of sorting and the differences between the sorting algorithms. They (the questions) should be between 5 and 10 and will include short answer questions and fill in the gaps.  **E-tivity 2**  Have a look at this [learning resource](https://betterprogramming.pub/5-basic-sorting-algorithms-you-must-know-9ef5b1f3949c) on sorting algorithms. It explains 5 sorting algorithms and provides sample algorithms for each of them. Convert each of the 5 algorithms provided into a C++ code. Use the online [C++ compiler](https://www.programiz.com/cpp-programming/online-compiler/) to run your code. Submit your code and the output result to the tutor through the email he/she will provide and by the deadline he specifies.  **E-tivity 3**  Identify and analyse/explain a real-life scenario where each of the five algorithms in E-tivity 2 can be applied. Send your answer to the tutor’s email by the deadline he/she will specify. | | |
| Where do they do it? | Online using their PCs and laptops. | | |
| By when should they do it? | The deadline for each E-tivity will be given by the tutor/e-moderator. | | |
| E-moderator/tutor role | | | |
| E-moderator will be expected to   * prepare the e-tivities * give you specific assistance/support (when needed) * ensure the learning materials are available and accessible * assess the e-tivities | | | |
| How are the learning outcomes in this unit assessed? | | Number of hours | 1 |
| The tutor/e-moderator will prepare a Rubric to assess E-tivities 1 and 3. E-tivity can be assessed with this [Rubric](https://drive.google.com/file/d/1nzUXDM5fZAIi3c78hCNvyIwIRdKESoK4/view?usp=sharing). | | | |
| How does this section link to other sections of the module? | | | |
| This Unit links directly to Units 10 and 11. Unit 10 and 11 are designed to teach trees which are sort algorithms based on Binary Search Tree data structure. | | | |

|  |  |
| --- | --- |
| = Total number of hours | 6 |

|  |  |
| --- | --- |
| **Some important questions** | |
| Which learning resources/ references will scaffold the students’ learning? | The following resources on Tree have been selected to aid your learning of the unit.  The resources include   1. [Sorting Algorithms in Data Structures](https://www.simplilearn.com/sorting-algorithms-in-data-structure-free-course-skillup) 2. [5-basic-sorting-algorithms-you-must-know](https://betterprogramming.pub/5-basic-sorting-algorithms-you-must-know-9ef5b1f3949c) 3. Sorting, Sets, and Selection. Pages 499 - 544 of [Data Structures and Algorithms in C++](http://160592857366.free.fr/joe/ebooks/ShareData/Data%20Structures%20and%20Algorithms%20in%20C++%202e%20By%20Michael%20Goodrich,%20Roberto%20Tamassia%20and%20David%20Mount.pdf). 4. [Data Structures and Algorithms Made Easy](http://www.dhimangaurav.com/docs/data.pdf) - Page 504 - 546 |
| How are students enabled to access the resources? | Students are provided with internet service on campus and in university hostels. They are provided with links to the videos and readings. They can watch the videos and read the book online or they can download it for offline use. Students can print the section on Sorting Algorithms from the reading resource. |
| Where in this unit are students expected to work collaboratively? | N/A |
| How has an inclusive approach been incorporated in this unit? | Different versions of the learning resources are provided - PDF and video versions to cater to students of different learning needs |
| How will feedback on unit be obtained from students? | As part of their unit’s E-tivity results submission, students are encouraged to add a note of any observations/concerns they have about the e-tivities. Together with student performance in the e-tivities, they will serve as feedback for the unit. |
| How will student feedback be used to improve unit? | Feedback will be forwarded to the course development team for consideration and implementation where appropriate. |
| At which point(s) will students receive formative feedback on the work they have done in the unit? | Students will receive feedback before the start of Week 9. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit-level overview** | | **Unit** |  |
| Topic name: | **Searching Algorithms** | | |
| Aim of the topic: | To teach you how search algorithms are designed and implemented as well as the science and concepts behind this data structure. | | |
| This topic covers: | The theory, practice, and application of various search algorithms. The theory will introduce you to the concept and principle of sorting and the differences between the various types of search algorithms. The practice and implementation will delve into writing search algorithms and implementing them in C++ as well as identifying and analysing situations where sorting will be applicable. | | |
| Intended learning outcomes: | *At the end of this* ***topic****, you will be able to:*   1. describe and differentiate between the types of search algorithms 2. implement a search algorithm in C++ 3. identify real-world problems that can be solved by search algorithms | | |

|  |  |
| --- | --- |
| Overview of student activity: | In this Unit, you will be engaged in the following activities designed to achieve the unit’s learning outcomes. They include   1. attending an online lecture 2. performing E-tivity 1, 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Constructive alignment of unit level outcomes with module level outcomes, learning activities and assessment** *(Pressing <Tab> at the end of the table will provide additional rows in the table, if required.)* | | | |
| Intended unit learning outcomes: | No of module-level outcome | Activity where students engage with this outcome | Where and how is this outcome assessed? |
| ***At the end of this unit, you will be able to:*** | | | |
| 1. describe and differentiate between the types of search algorithms | 1 | E-tivity 2 | The tutor will use the E-tivity Rubric for an online assessment |
| 1. implement a search algorithm in C++ | 2 | E-tivity 1 | The tutor will use the E-tivity Rubric for an online assessment |
| 1. identify real-world problems that can solved by search algorithms | 3 | E-tivity 2 | The tutor will use the E-tivity Rubric for an online assessment |

|  |  |  |  |
| --- | --- | --- | --- |
| Detailed explanation of ALL student and teacher engagement with the unit:  ***(This should be presented in the order that the activities take place. So if students do work* online *before* *coming to the lecture, that should be shown ahead of what happens in class.***  ***If there is more than one opportunity for face-to-face contact, or more than one online task, there should be a separate section for each instance, and they should be presented in the template in the same order that students encounter them.)***  ***Content*** *– such as lecture material – can EITHER be shown here OR added as* ***clearly identifiable*** *addenda to the document. If you plan to use addenda, you should ensure that these are cross-referenced in this section.)* | | | |
| Module-level outcomes addressed: | | | |
| Module-level outcomes 1, 2 and 3 | | | |
| Purpose of the unit: | | | |
| To add to your analytical and programming skills the ability to effectively and efficiently use search algorithms in your programs. | | | |
| Over to you: *(a description of the process of the section)* | | | |
| Search algorithms are described as the best tool to retrieve pieces of information from a data structure and are widely used in most programs, especially in commercial search engines. In this unit, you will learn all about search algorithms and their implementation in C++. You will attend an online lecture and do two E-tivities that will help you achieve the unit learning outcomes. Learning resources have been carefully selected to aid your learning. | | | |
| Pre-topic activity: | | Number of hours |  |
| N/A | | | |
| Face to face time: *(if applicable)* | | Number of hours |  |
| N/A | | | |
| Online activity: | | Number of hours | 5 |
| What should students do? | **Online Lecture**  You will attend an online lecture via Google Meet or Zoom. In the lecture you will learn about the concept and importance of searching algorithms, the types of search algorithms, how to write searching algorithms, and writing C++ code to implement searching algorithms. The following types of search algorithms will be discussed.   * Linear search * Binary search * Jump search * Sublist search * Interpolation search * Fibonacci search   **E-tivity 1**  Have a look at this [learning resource](https://codecoda.com/en/blog/entry/search-algorithms-and-their-implementations). It explains the types of searching algorithms and their implementation. Convert each of the algorithms provided into a C++ code. Use the online C++ [compiler](https://www.programiz.com/cpp-programming/online-compiler/) to run your code. Submit your code and the output result to the tutor through the email he/she will provide and by the deadline he specifies.  **E-tivity 2**  Briefly differentiate between linear and binary search algorithms and identify a real-world problem scenario where each of the two algorithms in E-tivity 1 can be applied. Send your answer to the tutor’s email by the deadline he/she will specify. | | |
| Where do they do it? | Online via Zoom or Google Meet. You will need your laptop or personal computer also for this unit. | | |
| By when should they do it? | The deadline for completion of the unit and submission of E-tivity results will be specified by the tutor/e-moderator. | | |
| E-moderator/tutor role | | | |
| The e-moderator/tutor will be expected to   * give the online lecture * prepare the e-tivities * give you specific assistance/support (when needed) * ensure the learning materials are available and accessible * assess the e-tivities | | | |
| How are the learning outcomes in this unit assessed? | | Number of hours | 1 |
| The tutor will prepare a Rubric to assess E-tivity 1 and can use this [Rubric](https://drive.google.com/file/d/1-MuUbJl1xOr3oWWWxlUEtwyxcFoT8H_q/view?usp=sharing) to assess E-tivity 2. | | | |
| How does this section link to other sections of the module? | | | |
| This Unit links directly to Units 10 and 11. Units 10 and 11 are designed to teach trees that utilise search algorithms when implementing Binary Search Tree data structures. | | | |

|  |  |
| --- | --- |
| = Total number of hours | 6 |

|  |  |
| --- | --- |
| **Some important questions** | |
| Which learning resources/ references will scaffold the students’ learning? | The following resources on Tree have been selected to aid your learning of the unit.  The resources include   1. [Sorting Algorithms in Data Structures](https://www.simplilearn.com/sorting-algorithms-in-data-structure-free-course-skillup) 2. [Let’s Understanding Searching Algorithms](https://codersera.com/blog/let-us-understand-searching-algorithms/) 3. [Search Algorithms and their Implementation](https://codecoda.com/en/blog/entry/search-algorithms-and-their-implementations) 4. [Introduction to Searching Algorithms in C++](https://www.softwaretestinghelp.com/searching-algorithms-in-cpp/) 5. [Data Structures and Algorithms Made Easy](http://www.dhimangaurav.com/docs/data.pdf) - Page 547 - 589 |
| How are students enabled to access the resources? | Students are provided with internet service on campus and in university hostels. They are provided with links to the videos and readings. They can watch the videos and read the book online or they can download it for offline use. Students can print the section on Searching Algorithms of the reading resource. |
| Where in this unit are students expected to work collaboratively? | N/A |
| How has an inclusive approach been incorporated in this unit? | Different versions of the learning resources are provided - PDF and video versions to cater to students with different learning needs. |
| How will feedback on unit be obtained from students? | As part of their unit’s E-tivity results submission, students are encouraged to add a note of any observations/concerns they have about the e-tivities. Together with student performance in the e-tivities, they will serve as feedback for the unit. |
| How will student feedback be used to improve unit? | Feedback will be forwarded to the course development team for consideration and implementation where appropriate. |
| At which point(s) will students receive formative feedback on the work they have done in the unit? | You will receive feedback before the start of Week 10. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit-level overview** | | **Unit** |  |
| Topic name: | **Trees I** | | |
| Aim of the topic: | This unit is designed to introduce you to tree data structures and their relevance in program and algorithm design. | | |
| This topic covers: | * The theoretical aspects of tree data structures. Introduction to tree data structures, types of tree data structures, operations of tree data structures, etc. | | |
| Intended learning outcomes: | *At the end of this* ***topic****, you will be able to:*   1. describe tree data structures 2. explain and differentiate between the types of tree data structures 3. explain the importance of tree data structures in programming | | |

|  |  |
| --- | --- |
| Overview of student activity: | You will be engaged in the following activities designed to achieve the unit’s learning outcomes. They include   1. a pre-topic activity 2. attending a face-to-face lecture 3. performing E-tivity 1 and 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Constructive alignment of unit level outcomes with module level outcomes, learning activities and assessment** *(Pressing <Tab> at the end of the table will provide additional rows in the table, if required.)* | | | |
| Intended unit learning outcomes: | No of module-level outcome | Activity where students engage with this outcome | Where and how is this outcome assessed? |
| ***At the end of this unit, you will be able to:*** | | | |
| 1. describe tree data structures | 1 | E-tivity 1 | The tutor will design a rubric to assess this E-tivity online. |
| 1. explain and different between the types of tree data structures | 1 | E-tivity 1 | The tutor will design a rubric to assess this E-tivity online. |
| 1. explain the importance of tree data structures in programming | 1 | E-tivity 2 | The tutor will design a rubric to assess this E-tivity online. |

|  |  |  |  |
| --- | --- | --- | --- |
| Detailed explanation of ALL student and teacher engagement with the unit:  ***(This should be presented in the order that the activities take place. So if students do work* online *before* *coming to the lecture, that should be shown ahead of what happens in class.***  ***If there is more than one opportunity for face-to-face contact, or more than one online task, there should be a separate section for each instance, and they should be presented in the template in the same order that students encounter them.)***  ***Content*** *– such as lecture material – can EITHER be shown here OR added as* ***clearly identifiable*** *addenda to the document. If you plan to use addenda, you should ensure that these are cross-referenced in this section.)* | | | |
| Module-level outcomes addressed: | | | |
| Module level outcome 1 | | | |
| Purpose of the unit: | | | |
| Trees are an essential data structure for storing hierarchical data with a directed flow. You will learn to visualise the tree data structure and when to use them and improve your knowledge of data structures and algorithms. | | | |
| Over to you: *(a description of the process of the section)* | | | |
| This unit will help you get a solid foundation in data structures and algorithms. You will get to understand why tree data structures are important, the structure of tree data structures, and the types of trees. There is a pre-topic activity where you will read (from the link provided in the pre-topic activity section) and write short notes on trees. Later, you will perform two e-tivities which are detailed in the online activity section. | | | |
| Pre-topic activity: | | Number of hours | 1 |
| Read introduction to [Tree in Data Structure](https://www.simplilearn.com/tutorials/data-structure-tutorial/trees-in-data-structure) and make short notes on Trees. You will present your notes in the face-to-face class. | | | |
| Face to face time: *(if applicable)* | | Number of hours |  |
| N/A | | | |
| Online activity: | | Number of hours | 4 |
| What should students do? | **Online Lecture**  Attend a two-hour online lecture designed to help you   1. understand the **difference** in the **structure** of a tree and linked list 2. some basic terminologies such as root, node, left child, right child, n-array tree, **binary** tree 3. perform **insertion**, **deletion**, and **search** along with their **time complexities**   **E-tivity 1**  Answer the [Multiple Choice Questions](https://home.cs.colorado.edu/~main/questions/chap10q.html) section of these questions and submit your answer to the tutor’s email.  **E-tivity 2**   1. Explain the importance of tree data structures in programming and algorithm design 2. Draw a binary search tree by insertion node by sequence for the following 3. C, O, M, P, U, T, E 4. e, b, d, f, a, g, c | | |
| Where do they do it? | You will do this assignment on your laptop or personal computers | | |
| By when should they do it? | The e-moderator will decide on a deadline and communicate to you. | | |
| E-moderator/tutor role | | | |
| E-moderator will   * prepare the e-tivities * give you specific assistance/support (when needed) * ensure the learning materials are available and accessible * assess the e-tivities | | | |
| How are the learning outcomes in this unit assessed? | | Number of hours | 1 |
| The tutor will prepare answer keys to assess E-tivity 1 and a Rubric for E-tivity 2. The tutor could use this [Rubric](https://drive.google.com/file/d/1exs0sCCmhLmImecUotl_Tlszv_6Si_GC/view?usp=sharing) as a guide for preparing the E-tivity 2 Rubric | | | |
| How does this section link to other sections of the module? | | | |
| This unit links to Unit 11 of the module. Unit 11 is a continuation of Unit 10. | | | |

|  |  |
| --- | --- |
| = Total number of hours | 6 |

|  |  |
| --- | --- |
| **Some important questions** | |
| Which learning resources/ references will scaffold the students’ learning? | The following resources on Tree have been selected to aid your learning of the unit.  The resources include   1. [An Introduction to Tree in Data Structure](https://www.simplilearn.com/tutorials/data-structure-tutorial/trees-in-data-structure) 2. Trees - Pages 267 - 320 of the book [Data Structures and Algorithms in C++](http://160592857366.free.fr/joe/ebooks/ShareData/Data%20Structures%20and%20Algorithms%20in%20C++%202e%20By%20Michael%20Goodrich,%20Roberto%20Tamassia%20and%20David%20Mount.pdf). |
| How are students enabled to access the resources? | Students are provided with internet service on campus and in university hostels. They are provided with links to the videos and readings. They can watch the videos and read the book online or they can download it for offline use. Students can print the section of Trees of the reading resource. |
| Where in this unit are students expected to work collaboratively? | N/A |
| How has an inclusive approach been incorporated in this unit? | Different versions of the learning resources are provided - PDF and video versions to cater to students with different learning needs. |
| How will feedback on unit be obtained from students? | As part of their unit’s E-tivity results submission, students are encouraged to add a note of any observations/concerns they have about the e-tivities. Together with student performance in the e-tivities, they will serve as feedback for the unit. |
| How will student feedback be used to improve unit? | Feedback will be forwarded to the course development team for consideration and implementation where appropriate. |
| At which point(s) will students receive formative feedback on the work they have done in the unit? | You will receive feedback before the start of Week 11. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit-level overview** | | **Unit** |  |
| Topic name: | **Trees II** | | |
| Aim of the topic: | To get a deeper understanding of tree data structures and their application in solving real-world problems. | | |
| This topic covers: | * The practical implementation of tree data structures in various scenarios to solve real-world problems. The application will include binary tree search, spanning trees, syntax tree, and heap. | | |
| Intended learning outcomes: | *At the end of this* ***topic****, you will be able to:*   1. identify application areas of trees in programming requirements 2. write algorithms and C++ codes to implement diverse types of trees | | |

|  |  |
| --- | --- |
| Overview of student activity: | You will be engaged in the following activities designed to achieve the unit’s learning outcomes. They include   1. attending a face-to-face lecture 2. performing E-tivity 1 and 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Constructive alignment of unit level outcomes with module level outcomes, learning activities and assessment** *(Pressing <Tab> at the end of the table will provide additional rows in the table, if required.)* | | | |
| Intended unit learning outcomes: | No of module-level outcome | Activity where students engage with this outcome | Where and how is this outcome assessed? |
| ***At the end of this unit, you will be able to:*** | | | |
| 1. Identify application areas of trees in programming requirements | 2 | E-tivity 1 | Online assessment with E-tivity 1 Rubric |
| 1. Write algorithms and C++ codes to Implement diverse types of trees | 3 | E-tivity 2 | Online assessment with E-tivity 2 Rubric on LMS |

|  |  |  |  |
| --- | --- | --- | --- |
| Detailed explanation of ALL student and teacher engagement with the unit:  ***(This should be presented in the order that the activities take place. So if students do work* online *before* *coming to the lecture, that should be shown ahead of what happens in class.***  ***If there is more than one opportunity for face-to-face contact, or more than one online task, there should be a separate section for each instance, and they should be presented in the template in the same order that students encounter them.)***  ***Content*** *– such as lecture material – can EITHER be shown here OR added as* ***clearly identifiable*** *addenda to the document. If you plan to use addenda, you should ensure that these are cross-referenced in this section.)* | | | |
| Module-level outcomes addressed: | | | |
| Module-level outcomes 1, 2, 3 | | | |
| Purpose of the unit: | | | |
| The purpose of the unit is to give you the ability to identify and apply trees in solving real-world problems. | | | |
| Over to you: *(a description of the process of the section)* | | | |
| This unit is a continuation of Unit 9. It is designed to help you identify and apply tree data structures to solve real-world problems. There are two e-tivities designed to achieve this purpose and an online lecture. E-tivity will test your knowledge on trees and E-tivity 2 will help you put your analytical, critical thinking, and code writing skills into action. | | | |
| Pre-topic activity: | | Number of hours |  |
| N/A | | | |
| Face to face time: *(if applicable)* | | Number of hours |  |
| N/A | | | |
| Online activity: | | Number of hours | 5 |
| What should students do? | **Online Lecture**  You will attend a two-hour online lecture designed to help you   1. understand tree traversal Algorithms BFS, DFS, Level order 2. understand Binary search tree 3. understand how to do pre-order, post-order, and in-order traversals 4. apply trees in different real-world scenarios to solve problems   **E-tivity 1**  The following are some instances where tree data structures can be applied   1. When the information you have to store has relationships between parts that look like a tree i.e., when there is a natural tree structure to the data, e.g., genealogy, a file system with directories. 2. When you need to perform operations on the information for which a tree is advantageous. For example, a binary search algorithm on a b-tree. 3. When the information accumulates over time in a natural tree structure. 4. When you need to output information in a tree, depth-first or breadth-first.   Identify and draw the structure of the type of tree which can be applied in the instances described above. Submit a PDF file of your completed work to the tutor on the specified deadline.  **E-tivity 2**  Write C++ code to illustrate the following tree operations: pre-order, post-order, and in-order traversals. Compile your code with the [online compiler](https://www.programiz.com/cpp-programming/online-compiler/) and submit the code and results to the Unit section on the LMS. | | |
| Where do they do it? | Online via Zoom or Google Meet and your laptops and personal computers and the LMS. | | |
| By when should they do it? | The e-moderator will decide on a deadline and communicate to you. | | |
| E-moderator/tutor role | | | |
| E-moderator will   * prepare the e-tivities * give you specific assistance/support (when needed) * ensure the learning materials are available and accessible * assess the e-tivities | | | |
| How are the learning outcomes in this unit assessed? | | Number of hours | 1 |
| E-tivity 1 and 2 will be assessed with this [Rubric](https://drive.google.com/file/d/1P7qgIomequRu6RRzdqjLR3PNN6-Yvht9/view?usp=sharing). | | | |
| How does this section link to other sections of the module? | | | |
| This unit links to Units 10 and 12. Unit 11 is a direct continuation of this Unit and Unit 12. Units 10, 11, and 12 all discuss non-linear data structures. | | | |

|  |  |
| --- | --- |
| = Total number of hours | 6 |

|  |  |
| --- | --- |
| **Some important questions** | |
| Which learning resources/ references will scaffold the students’ learning? | The following resources on Tree have been selected to aid your learning of the unit.  The resources include   1. [An Introduction to Tree in Data Structure](https://www.simplilearn.com/tutorials/data-structure-tutorial/trees-in-data-structure) 2. Trees - Pages 267 - 320 of the book [Data Structures and Algorithms in C++](http://160592857366.free.fr/joe/ebooks/ShareData/Data%20Structures%20and%20Algorithms%20in%20C++%202e%20By%20Michael%20Goodrich,%20Roberto%20Tamassia%20and%20David%20Mount.pdf). 3. Binary Search Trees on Pages 286 - 307 of [Introduction to Algorithms](https://edutechlearners.com/download/Introduction_to_algorithms-3rd%20Edition.pdf) 4. [How to Implement Binary Tree in Data Structure](https://www.simplilearn.com/tutorials/data-structure-tutorial/binary-tree-in-data-structures) 5. [Heap Data Structures](https://cppsecrets.com/users/1353109971109711511910511010548575764103109971051084699111109/C00-Program-to-Implement-Heap.php) |
| How are students enabled to access the resources? | The university has provided internet service on campus and the university hostels for all students. They are also provided internet link to all online learning resources. Students can watch the videos and read the books online or they can download for offline use. Students can print the section of Trees of the reading resource. |
| Where in this unit are students expected to work collaboratively? | N/A |
| How has an inclusive approach been incorporated in this unit? | Different versions of the learning resources are provided - PDF and video versions to cater for students of different learning needs. |
| How will feedback on unit be obtained from students? | As part of their unit’s E-tivity results submission, students are encouraged to add a note of any observations/concerns they have about the e-tivities. Together with student performance in the e-tivities, they will serve as feedback for the unit. |
| How will student feedback be used to improve unit? | Feedback will be forwarded to the course development team for consideration and implementation where appropriate. |
| At which point(s) will students receive formative feedback on the work they have done in the unit? | Students will receive feedback before the start of Week 12. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit-level overview** | | **Unit** |  |
| Topic name: | **Graphs** | | |
| Aim of the topic: | Understand the science and concept of Graph data structures and how they can be used to address real-world problems in which it represents the problem area as a network like telephone networks, circuit networks, and social networks. | | |
| This topic covers: | The following topics will be covered in the unit   * Introduction to Graphs * Graph terminology * Graph representation * Graph operations * Real-world graph application | | |
| Intended learning outcomes: | *At the end of this* ***topic****, you will be able to:*   1. describe graph data structures 2. differentiate between the various graph data structures 3. write algorithms for graph data structures and their implementation in C++ 4. apply graphs to solve real-world problems | | |

|  |  |
| --- | --- |
| Overview of student activity: | In this Unit, you will be engaged in the following activities designed to achieve the unit’s learning outcomes. They include   1. attending an online lecture 2. performing E-tivity 1, 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Constructive alignment of unit level outcomes with module level outcomes, learning activities and assessment** *(Pressing <Tab> at the end of the table will provide additional rows in the table, if required.)* | | | |
| Intended unit learning outcomes: | No of module-level outcome | Activity where students engage with this outcome | Where and how is this outcome assessed? |
| ***At the end of this unit, you will be able to:*** | | | |
| 1. describe graph data structures | 1 | E-tivity 1 | Assessment will be done online with the E-tivity 1 Rubric |
| 1. differentiate between the various graph data structures | 1 | E-tivity 1 | Assessment will be done online with the E-tivity 1 Rubric |
| 1. write algorithms for graph data structures and its implementation in C++ | 2 | E-tivity 2 | Assessment will be done online with the E-tivity 2 Rubric |
| 1. apply graphs to solve real world problems | 2 | E-tivity 2 | Assessment will be done online with the E-tivity 2 Rubric |

|  |  |  |  |
| --- | --- | --- | --- |
| Detailed explanation of ALL student and teacher engagement with the unit:  ***(This should be presented in the order that the activities take place. So if students do work* online *before* *coming to the lecture, that should be shown ahead of what happens in class.***  ***If there is more than one opportunity for face-to-face contact, or more than one online task, there should be a separate section for each instance, and they should be presented in the template in the same order that students encounter them.)***  ***Content*** *– such as lecture material – can EITHER be shown here OR added as* ***clearly identifiable*** *addenda to the document. If you plan to use addenda, you should ensure that these are cross-referenced in this section.)* | | | |
| Module-level outcomes addressed: | | | |
| Module-level outcomes 1, 2 and 3 | | | |
| Purpose of the unit: | | | |
| Graphs are considered the perfect data structure for modelling networks, which makes them an indispensable piece of your data structure toolkit. This unit has been designed to equip you with the knowledge and skills needed to use Graphs data structures effectively and efficiently in your software engineering tasks. | | | |
| Over to you: *(a description of the process of the section)* | | | |
| Graphs are a powerful and versatile data structure that easily allows you to represent real-life relationships between different types of data (nodes). They are composed of nodes, or *vertices*, which hold data, and *edges*, which are a connection between two vertices. A single node is a *vertex*. Consider a map of the area where you live. As a graph, we could model bus stops as vertices, with bus routes between stops functioning as the edges. What about the internet? Web pages can be vertices, and the hyperlinks which connect them are edges. Real-world relationships modeled as graphs are numerous, making them an essential concept to master. At the end of this unit, you will understand how to apply graphs in all of these situations. You will attend an online lecture and perform two E-tivities designed to help you achieve the unit learning outcomes. | | | |
| Pre-topic activity: | | Number of hours |  |
| N/A | | | |
| Face to face time: *(if applicable)* | | Number of hours |  |
| N/A | | | |
| Online activity: | | Number of hours | 5 |
| What should students do? | **Online Lecture**  You will attend an online lecture via Google Meet or Zoom. In the lecture, you will learn about the concept and importance of tree data structures, the types of tree data structures, and how to write tree algorithms and implement them in C++. The lecture will also discuss   * Introduction to Graphs * Graph terminology * Graph representation * Graph operations * Real-world graph application   **E-tivity 1**  With the aid of a diagram, briefly describe and differentiate between the following types of graphs: Finite Graph, Infinite Graph, Trivial Graph, Simple Graph, and Multigraph. Submit your response to the tutor’s email by the deadline he/she will specify.  **E-tivity 2**  Describe how you will use a graph data structure to build a social network. Provide the algorithm for the network and its implementation in C++. Run your code using the [online compiler](https://www.programiz.com/cpp-programming/online-compiler/). Submit your code and the output result to the tutor through the email he/she will provide and by the deadline he specifies. | | |
| Where do they do it? | Online via Zoom or Google Meet. You will need your laptop or personal computer for this unit. | | |
| By when should they do it? | During the revision week | | |
| E-moderator/tutor role | | | |
| E-moderator will   * prepare the e-tivities * give you specific assistance/support (when needed) * ensure the learning materials are available and accessible * assess the e-tivities | | | |
| How are the learning outcomes in this unit assessed? | | Number of hours | 1 |
| The tutor will prepare a Rubric to assess E-tivity 1. E-tivity 2 can be assessed with this [Rubric](https://drive.google.com/file/d/1rPtkHBqwL1ivxC5E32pwMyWXSL4pjaiv/view?usp=sharing). | | | |
| How does this section link to other sections of the module? | | | |
| This unit links to Units 10 and 11 due to the similarities between the data structures discussed in these units. All three units - 10, 11, and 12 discuss non-linear data structures. | | | |

|  |  |
| --- | --- |
| = Total number of hours | 6 |

|  |  |
| --- | --- |
| **Some important questions** | |
| Which learning resources/ references will scaffold the students’ learning? | The following resources on Trees have been selected to aid your learning of the unit.  The resources include   1. [Graph Data Structure](https://www.programiz.com/dsa/graph) 2. [Your One-Stop Solution for Graphs in Data Structures](https://www.simplilearn.com/tutorials/data-structure-tutorial/graphs-in-data-structure) 3. [A Gentle Introduction to Data Structures: How Graphs Work](https://www.freecodecamp.org/news/a-gentle-introduction-to-data-structures-how-graphs-work-a223d9ef8837/) 4. [Data Structures and Algorithms Made Easy](http://www.dhimangaurav.com/docs/data.pdf) - Page 426 - 503 5. [Graph Algorithms](http://160592857366.free.fr/joe/ebooks/ShareData/Data%20Structures%20and%20Algorithms%20in%20C++%202e%20By%20Michael%20Goodrich,%20Roberto%20Tamassia%20and%20David%20Mount.pdf) - Pages 593 - 653 |
| How are students enabled to access the resources? | Students are provided with internet service on campus and in university hostels. They are provided with links to the videos and readings. They can watch the videos and read the book online or they can download it for offline use. Students can print the section of Graphs of the reading resource. |
| Where in this unit are students expected to work collaboratively? | N/A |
| How has an inclusive approach been incorporated in this unit? | Different versions of the learning resources are provided - PDF and video versions to cater for students of different learning needs. |
| How will feedback on unit be obtained from students? | Students will be given the opportunity to evaluate the entire module as well as assess the tutor based on the criteria set by the university’s Quality Assurance Unit. The university considers the assessment of tutors a criterion for promotion and evaluation a requisite for improving and redesigning the module. |
| How will student feedback be used to improve unit? | Feedback will be forwarded to the course development team for consideration and implementation where appropriate. |
| At which point(s) will students receive formative feedback on the work they have done in the unit? | You will receive feedback before the start of the semester examinations. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit-level overview** | | **Unit** | **13 & 14** |
| Topic name: | **Revision and End of Semester Examinations** | | |
| Aim of the topic: | To test the understanding, recall, analysis, and application of lessons in Units 1 - 12 of the module. | | |
| This topic covers: | A revision of Units 1 - 12 and a cumulative assessment test on major aspects of the Units 1 - 12 of the module. | | |
| Intended learning outcomes: | *At the end of this* ***topic****, you will be able to:*   1. personally assess how far or close you are from achieving the module learning outcomes | | |

|  |  |
| --- | --- |
| Overview of student activity: | You will attend an online revision class and write a two-hour cumulative assessment exam covering major aspects of Units 1 - 12. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Constructive alignment of unit level outcomes with module level outcomes, learning activities and assessment** *(Pressing <Tab> at the end of the table will provide additional rows in the table, if required.)* | | | |
| Intended unit learning outcomes: | No of module-level outcome | Activity where students engage with this outcome | Where and how is this outcome assessed? |
| ***At the end of this unit, you will be able to:*** | | | |
| 1. assess how far or close you are from achieving the module learning outcomes | 1, 2 and 3 | End of semester examination | Assessment will be conducted offline using an end of semester examination assessment Rubric |

|  |  |  |  |
| --- | --- | --- | --- |
| Detailed explanation of ALL student and teacher engagement with the unit:  ***(This should be presented in the order that the activities take place. So if students do work* online *before* *coming to the lecture, that should be shown ahead of what happens in class.***  ***If there is more than one opportunity for face-to-face contact, or more than one online task, there should be a separate section for each instance, and they should be presented in the template in the same order that students encounter them.)***  ***Content*** *– such as lecture material – can EITHER be shown here OR added as* ***clearly identifiable*** *addenda to the document. If you plan to use addenda, you should ensure that these are cross-referenced in this section.)* | | | |
| Module-level outcomes addressed: | | | |
| Module-level outcomes 1, 2, and 3. | | | |
| Purpose of the unit: | | | |
| Assess your performance and ability to achieve the module and unit-level learning outcomes, allow tutors to assess if the module-level learning outcomes were achieved, and plan for the module delivery in the next academic year. | | | |
| Over to you: *(a description of the process of the section)* | | | |
| Revise all the units you have covered in class as well as all the activities in which you have engaged. Get a grasp of all the major aspects of the unit lessons to be in good standing for the assessment. Arrive at the examination hall on time and adhere to all examination rules and regulations during the exams. Good luck. | | | |
| Pre-topic activity: | | Number of hours | 3 |
| Revision of Units 1 -12 lessons and learning activities. | | | |
| Face to face time: *(if applicable)* | | Number of hours | 2 |
| A two-hour face-to-face examination. | | | |
| Online activity: | | Number of hours |  |
| What should students do? | N/A | | |
| Where do they do it? | N/A | | |
| By when should they do it? | N/A | | |
| E-moderator/tutor role | | | |
| Assist you with revision if needed and help prepare you for the examination. You may receive some trial questions from him/her and some useful examination tips and guidelines. | | | |
| How are the learning outcomes in this unit assessed? | | Number of hours | 1 |
| The tutor will design an appropriate marking scheme to grade the examination.  However, pay attention to the following   * demonstrate clarity of thought and expression in your writing * produce scripts with no to minimal spelling and grammar errors * organise your work properly with appropriate numbering and indentations * avoid too many cancellations | | | |
| How does this section link to other sections of the module? | | | |
| It tests the knowledge, understanding, application, and analysis abilities you have gained throughout all the Units in the module. It is the conclusion of the module. | | | |

|  |  |
| --- | --- |
| = Total number of hours | 6 |

|  |  |
| --- | --- |
| **Some important questions** | |
| Which learning resources/ references will scaffold the students’ learning? | You will utilise all the reading texts and video resources used for Units 1 - 12 to you in preparation for the end-of-semester exams. |
| How are students enabled to access the resources? | All reading and video learning resources for Units 1 to 12 have been made available to you. |
| Where in this unit are students expected to work collaboratively? | N/A |
| How has an inclusive approach been incorporated in this unit? | Provision will be made for students with any form of disability to write the examination in an inclusive environment. |
| How will feedback on unit be obtained from students? | Your performance at the end of the semester exam will be the main source of feedback for this unit. You are also allowed to give information on any personal observations you made during the exam to the tutor. |
| How will student feedback be used to improve unit? | Results and personal feedback from students will be analysed. Observed trends and insights will be a source of feedback for implementation where applicable to improve the module in the following year. |
| At which point(s) will students receive formative feedback on the work they have done in the unit? | As soon as practicable. Preferably, one month after the end of semester examinations. |

END OF UNIT