Question 4

Title: Paradigms of programming languages.

Description

Programming paradigms define the different approaches to problem solving during system analysis, design and implementation, as adopted by the several programming languages in use today. In this project, a number of such programming paradigms are studied and analysed with the scope of understanding the core concepts on which each paradigm is formulated, clarify the relationship between them and identifying the most appropriate applications per paradigm.

Instructions

This study must be presented in a literature review document. The document length should approximately 6000 words (approx 10-12 A4 pages). The document structure as defined in Appendix A must be adhered to, including fonts, line spacing and sectioning (supplementary sections may be added where deemed fit). Programming paradigms included in the study must conform to the following list. The core study is to be carried out, but not limited to, all of the given suggested papers.

Paradigms and suggested papers

Imperative Programming


Wirth, N. Program development by stepwise refinement. Comm. ACM 14, 221-227, Apr.1971

Declarative Programming with Logic Programming Languages

Declarative Programming with Functional Programming


Object-oriented Programming


Concurrent Programming Paradigm


Marking Scheme

- Document (100%)
  - Overall logical and formatting presentation (20%)
  - Programming Paradigms section (25%)
  - Discussion section (30%)
  - Conclusion (15%)
  - Bibliography + Referencing (10%)

Appendices

Appendix A defines the document structure to be adhered to when writing the literature review. Appendix B presents the citation guidelines, and a plagiarism information section is included in Appendix C.
Abstract

The abstract should enable potential readers of the document to identify the basic content of the document quickly and accurately, allowing them to determine its relevance to their interest, and so deciding whether to read the document in its entirety. Therefore the abstract should concisely state the principal objectives and scope of your study. More importantly, it should summarize the principal discussion points (e.g. The list of paradigms under considerations and two-three striking discussion points) and conclusions.

The abstract must be concise, not exceeding 200 words, and should not contain any references to literature, tables, figures or acronyms.

1. Introduction

The introduction should contain information about programming languages and programming paradigms in general. The introduction should also state the scope and the value to be gained from this study.

2. Different programming paradigms

The aim of this section is that of showing complete understanding of each of the programming paradigms by presenting the main points in your own words from the assigned papers. For each paradigm, an explanation of the concept on which each paradigm is based, paradigm distinguishing features and techniques, design patterns, and sample example paradigm implementing programming language code. (Papers must be well referenced.)

2.1 Imperative Programming Paradigm
Imperative programming Paradigm text.

2.2 Declarative Programming applied in Logic Programming
Logic Programming text.

2.3 Declarative Programming applied in Functional Programming Languages
Functional Programming text.

2.4 Object-Oriented Programming Paradigm
Object-Oriented programming Paradigm text.
2.5 Concurrent Programming Paradigm
Concurrent programming paradigm text.

3. Discussion
The discussion section of the literature review should present an analysis based on the knowledge gained from the research in the previous section. Note that for this section further literature sources may be necessary, other than the suggested list of papers.

3.1 Comparison
In this section the various programming paradigms are compared on factors such as programming approach, programming/debugging ease, development tool support and delivered code quality.

3.2 Applications
This section outlines the real-world application in which each paradigm has been applied.

3.3 Multiple paradigm support
Several programming languages in use today support two or more of the above paradigms. Give a description of 2 such languages and briefly underline the reason/effect behind the adoption of the multiple paradigms.

4. Conclusion
The conclusion should document in a concise manner the main lessons learned as a result of this study. (E.g. This study showed that specific paradigms are more suitable in certain scenarios…; The concepts on which these paradigms are based are…; ). The conclusion should not be longer than 250 words.

5. Bibliography
Bibliography entries order by surname using the Harvard citation style guide as defined in appendix B.
Appendix – B Guide to using the Harvard Citation System

Introduction

When writing reports, essays or dissertations it is important that you give your reader sufficient details to trace the references you use. This guide shows you how to use the Harvard Citation System, which is commonly used in academia because it is concise and makes citing other peoples work clearer to read and looks more professional.

Citations appear in the body of the text and references appear separately in a list called a Bibliography. This guide shows how to write citations and references. If you follow the examples in this guide your work will be correctly cited and referenced.

The importance of citing authors' work

Citing the authors you have consulted shows that you have researched your subject and where you got your information or arguments. It also acknowledges that previous authors have proposed arguments you raise. By not citing an author's work you may be accused of plagiarism (theft of ideas) which within Academia is considered as theft.

Citations

When you cite you should include the originator's name and the year of publication after each reference in your text. If the originator's name occurs naturally in the text, the year follows in brackets (parentheses). If it does not, then both name and year appear in brackets. You should also provide the page number wherever you can. This will help the reader when using your Bibliography.

Example

The problems of image retrieval are extenuated by the convergence of technology (Enser, 1995). This convergence has led to a number of intellectual property problems, noted by Horner et al (1994, p 231). It may be, as Gill (1989) pointed out, that we live in a techno-centric society which ignores the social needs of communities in the Information Society (Day, 1993; Leach, 1995).

References

References (Bibliographies) appear at the end of your text. Each reference has a number of parts which MUST be written in the correct order using the correct layout. The order of reference elements in the Harvard Systems are:

1. Author(s) name
2. Year of Publication
3. Titles of the publication
4. Place of publication and publisher

References

References (Bibliographies) appear at the end of your text. Each reference has a number of parts which MUST be written in the correct order using the correct layout. The order of reference elements in the Harvard Systems are:

1. Author(s) name
2. Year of Publication
3. Titles of the publication
4. Place of publication and publisher
The Harvard System has strict guidelines on how to present references in Bibliographies and this is as important as writing the reference correctly.

**Author's name**
The author's name(s) **must** be in capital letters. Surnames are written first followed by their initials. If there are two authors their order of appearance is exactly as they appear on the title page. If the source is from a corporate organisation then the author's name is the organisation. The name is followed by a comma (,).

**Year of Publication**
After the author's name write the year of publication followed by a full stop. If you cite two sources by the same author written in the same year you must differentiate the two by labelling them a or b etc. This must also be made clear in your citation.

**Title of the publication**
This must be written in italics or underlined. Only the first letter of the first word of the title needs to be in capital letters. Look at the examples below as journal articles and book chapters are referenced differently and it can get confusing!

**Place of publication**
The place is written followed by a colon (:) followed by the name of the publisher.

**Numeration**
For books this is the page numbers. For journals, this is the Volume, edition and page numbers.

**Location or type**
This is only used for non-print sources such as TV programmes, films, Websites or for rare documents.

### Books (monographs)

References must follow the order of elements. Use the following example as a guide:

**Example**

**Chapter in edited books**

If the source you are citing is written by one author and the book has been edited by other authors, you must treat the citation as being within a citation and the name of the book is written in italics (or underlined). Use pp.<starting page number> - <finishing page number> to indicate the range of pages.

**Example**


### Journal articles

The name of the Journal and **NOT** the name of the article should be written in italics (or underlined)

**Example**


Newspaper articles

Some newspaper articles will not have an author. If this is the case use the newspaper's name instead. You must put the date at the end of the citation.

Examples

Electronic documents (Websites)

The same format must be applied to websites as far as possible. Many websites do not give a publication date so do not guess when it was published. However, you must give the full web Uniform Resource Locator (URL) and the date when you looked at it.

Examples
http://www.ariadne.ac.uk/issue16/cover [Viewed 22-9-98].

Additional help

Hopefully, by following the examples in this guide you should be able to cite and reference your work correctly. If you need more help ask your lecturer or use some of the books about how to carry out research projects in the library.

Quotes

It is sometimes useful (but not essential) to quote an author's directly. The change in pace and style of writing often adds impact and draws the readers attention to an important point. However, don't be tempted to overdo the use of quotes. You do not need to quote every time you cite someone else's work. Separate quotes from the body of text by use of space lines before and after and indent the quote on both sides of the page.

Example

The proper way to quotes someone else's work is to indent the quote on both sides of the page and separate the quote from the body of the text by use of space lines. (Day, 1998, p.17)

Note also how the page number of the quote has been indicated for the quote, i.e. p.17. If a quote goes over a page then the correct form is pp.17-18. This is the same approach you would use for indicating chapters in books or articles in journals, newspapers, etc. and must always be used for quotes.
Appendix – C

Plagiarism: Information

To plagiarise is "to steal and pass off (the ideas or words of another) as one's own : use (another's production) without crediting the source" or "to commit literary theft : present as new and original an idea or product derived from an existing source" (Merriam-Webster Online).

The University of Malta defines plagiarism as the unacknowledged use, as one's own, of work of another person, whether or not such work has been published (University of Malta, Regulations Governing Conduct at Examinations, 1997 and Assessment Tests Regulations, 1997).

Plagiarism can be a deliberate intention to cheat, or it can be an innocent misuse of material without proper citation or acknowledgement. The Department of Computer Science and the Department of AI, the Faculty of ICT, and the University of Malta, will not tolerate plagiarism in any form. The minimum punishment is an appropriate reduction of marks. The maximum punishment, enforceable by Senate, is expulsion from the degree programme. It is the student's responsibility to ensure that the work presented for assessment is free from plagiarism.

To help students avoid plagiarism, the Department of Computer Science and the Department of AI jointly provide a short course each semester. Students are advised to attend as the Department assumes that all students know how to avoid plagiarism. A record of the details of an act of plagiarism may be permanently noted on a student's file.