

## Module 31 Cloud Services and Platforms

|   |   |
|---|---|
| <b>Module title</b>   | Cloud Services and Platforms  |
| <b>Module NFQ level (only if an NFQ level can be demonstrated)</b>  | 8   |
| <b>Module number/reference</b>  | BSCH-CSP  |
| <b>Parent programme(s)</b>  | Bachelor of Science (Honours) in Computing Science  |
| <b>Stage of parent programme</b>  | Award stage   |
| <b>Semester (semester1/semester2 if applicable)</b>   | Semester 2  |
| <b>Module credit units (FET/HET/ECTS)</b>   | ECTS  |
| <b>Module credit number of units</b>  | 5   |
| <b>List the teaching and learning modes</b>   | Direct, Blended   |
| <b>Entry requirements (statement of knowledge, skill and competence)</b>  | Learners must have achieved programme entry requirements.   |
| <b>Pre-requisite module titles</b>  | BSCH-CSWD, BSCH-SSWD  |
| <b>Co-requisite module titles</b>   | None  |
| <b>Is this a capstone module? (Yes or No)</b>   | No  |
| <b>Specification of the qualifications (academic, pedagogical and professional/occupational) and experience required of staff (staff includes workplace personnel who are responsible for learners such as apprentices, trainees and learners in clinical placements)</b> | Qualified to as least a Bachelor of Science (Honours) level in Computer Science or equivalent and with a Certificate in Training and Education (30 ECTS at level 9 on the NFQ) or equivalent. |
| <b>Maximum number of learners per centre (or instance of the module)</b>  | 60  |
| <b>Duration of the module</b>   | One Academic Semester, 12 weeks teaching  |
| <b>Average (over the duration of the module) of the contact hours per week</b>  | 3   |
| <b>Module-specific physical resources and support required per centre (or instance of the module)</b>   | One class room with capacity for 60 learners along with one computer lab with capacity for 25 learners for each group of 25 learners  |

| Analysis of required learning effort      |                                    |       |
|---|------------------------------------|-------|
|   | Minimum ratio<br>teacher / learner | Hours |
| <b>Effort while in contact with staff</b> |                                    |       |
| Classroom and demonstrations              | 1:60                               | 18    |
| Monitoring and small-group teaching       | 1:25                               | 18    |
| Other (specify)                           |                                    |       |
| <b>Independent Learning</b>               |                                    |       |
| Directed e-learning                       |                                    |       |
| Independent Learning                      |                                    | 44    |
| Other hours (worksheets and assignments)  |                                    | 45    |
| Work-based learning – learning effort     |                                    |       |
| <b>Total Effort</b>                       |                                    | 125   |

| Allocation of marks (within the module) |                          |                       |                                    |                                  |       |
|---|--------------------------|-----------------------|------------------------------------|----------------------------------|-------|
|   | Continuous<br>assessment | Supervised<br>project | Proctored practical<br>examination | Proctored written<br>examination | Total |
| <b>Percentage<br/>contribution</b>      | 50%                      |                       |                                    | 50%                              | 100%  |

### Module aims and objectives

The module aims to introduce the learner to cloud computing infrastructure. Learners learn state-of-the-art solutions offered by major cloud providers. Its main focus is on understanding the concepts and techniques which form the cloud infrastructure. A significant part of the assessment involves the learner developing a cloud application, deploying it to the Cloud and configuring for and evaluating performance post installation. Assessment emphasises a collaborative group approach to equip the learners with the skills required to work in a successful agile software development team.

### Minimum intended module learning outcomes

On successful completion of this module, the learner will be able to:

1. Analyse the goals, functions, models of cloud computing systems and understand the benefits of deploying applications on cloud infrastructure.
2. Compare the offerings of various cloud computing platform providers.
3. Illustrate the architecture of the Web and the Services Oriented Architectures.
4. Discuss the role of virtualisation in cloud computing.
5. Discuss existing techniques for cloud resource management.
6. Explain the techniques used in the construction of cloud storage.
7. Explore the security and privacy problems in cloud computing.

1. Develop and deploy applications over popular cloud computing infrastructures (e.g. Amazon, Azure, Google).

### **Rationale for inclusion of the module in the programme and its contribution to the overall MIPLOs**

The module enables learners to understand concepts which form the basis of cloud infrastructures. It enables the learners to develop and deploy applications on the commercially available cloud platform(s).

Appendix 1 of the programme document maps MIPLOs to the modules through which they are delivered.

### **Information provided to learners about the module**

Learners receive a programme handbook to include module descriptor, module learning outcomes (MIMLO), class plan, assignment briefs, assessment strategy, and reading materials.

### **Module content, organisation and structure**

#### **Basics of Cloud Computing**

- Cloud Computing Overview and History
  - Pre-Cloud history,
  - Cloud history and what challenges cloud computing has overcome,
  - Advantages and challenges, cloud popularity examples, Netflix, google, etc.
- Cloud Computing Deployment and Services Models
  - Service models – SaaS, PaaS, IaaS, XaaS
  - Deployment models – private, community, public, hybrid cloud
- Cloud Computing platforms (Amazon, Google, Microsoft Azure, VMWare, Open-Source, etc.)
  - Cloud infrastructure at Amazon,
  - Cloud infrastructure at Google,
  - Cloud infrastructure at Microsoft,
  - Open-Source software platforms for private clouds
- Web and Service Oriented Architecture(s) (SOAP, REST)
  - Web 1.0, 2.0, 3.0,
  - SOA concepts and its implementations (SOAP, RESTful)

## **Cloud Construction and Administration**

- Parallel and Distributed Computing
  - Parallel and Distributed Systems concepts:
  - Communication Protocols and Process Coordination,
  - Concurrency, Atomic Actions, etc.
- Cloud Resource Virtualization
  - Layering and Virtualization,
  - Cloud Virtualization, Virtual Machine Monitors and Virtual Machines,
  - Virtualization types: full virtualization para-virtualization
- Cloud Resource Management
  - Policies and Mechanisms for resource management,
  - Task scheduling on Cloud,
  - Scheduling algorithms
- Cloud Storage
  - Storage models,
  - Distributed File systems,
  - Google File system,
  - Apache Hadoop, etc.
- Cloud Security
  - Cloud security challenges,
  - Privacy impact assessment,
  - Operating system security, virtual machine security, security of virtualization

## **Cloud Programming**

- Cloud Programming Techniques
  - Optimisation of cloud applications,
  - Memcache, Sharding counters, Minimizing work, Map Reduce, etc.
- Cloud Application Development
  - Technical skills related to development, deployment, and configuration of cloud application for a chosen cloud platform.

## **Module teaching and learning (including formative assessment) strategy**

The module is taught using a combination of lectures, demonstrations, and tutorials. The demonstrations and tutorials focus on getting learners up to standard in practical application development. The lectures supply the necessary theoretical background. In a fast-changing technology field learners are expected under guidance to engage in research in relation to the different technologies and products available.

The module has both a continuous assessment element and a final examination. It requires the learner to show an understanding of the emerging technologies in

relation to the Web, the Web as Cloud and how to develop applications in that environment. The first component of assessment allows the learners to gain some experience in researching issues around Cloud Computing, the second gives them practical hands-on experience of building and deploying a Cloud application, and the final examination assesses their understanding to the theoretical issues involved.

### **Timetabling, learner effort and credit**

The module is timetabled as one 1.5-hour lecture and one 1.5-hour labs per week.

The number of 5 ECTS credits assigned to this module is our assessment of the amount of learner effort required. Continuous assessment spreads the learner effort to focus on small steps before integrating all steps into the overall process of developing and deploying a cloud application.

There are 36 contact hours made up of 12 lectures delivered over 12 weeks with classes taking place in a classroom. There are also 12 lab sessions delivered over 12 weeks taking place in a fully equipped computer lab. The learner will need 45 hours of independent effort to further develop the skills and knowledge gained through the contact hours. An additional 44 hours are set aside for learners to work on worksheets and assignments that must be completed for the module as a part of the continuous assessment.

The team believes that 125 hours of learner effort are required by learners to achieve the MIMLOs and justify the award of 5 ECTS credits at this stage of the programme.

### **Work-based learning and practice-placement**

There is no work based learning or practice placement involved in the module.

### **E-learning**

The college VLE is used to disseminate notes, advice, and online resources to support the learners. The learners are also given access to Lynda.com as a resource for reference.

### **Module physical resource requirements**

Requirements are for a classroom for 60 learners equipped with a projector, and a 25-seater computer lab for practical sessions with access to recommended cloud application development platform (this decision may be left on the lecturer to take before the start of the module).

## Reading lists and other information resources

### Recommended Reading

Marinescu, D. C. (2017) *Cloud Computing: Theory and Practice*. Cambridge MA: Morgan Kaufmann.

### Secondary Reading

Sanderson, D. (2015) *Programming Google App Engine with Java*. Sebastopol: O'Reilly.

Recent conference/journal papers on cloud computing (related to module topics) as well as documentation from cloud infrastructure providers.

Rhoton, J. (2013) *Cloud Computing Explained*. London: Recursive Press.

### Specifications for module staffing requirements

For each instance of the module, one lecturer qualified to at least Bachelor of Science (Honours) in Computer Science or equivalent, and with a Certificate in Training and Education (30 ECTS at level 9 on the NFQ) or equivalent.. Industry experience would be a benefit but is not a requirement.

Learners also benefit from the support of the programme director, programme administrator, learner representative and the Student Union and Counselling Service.

### Module Assessment Strategy

The assignments constitute the overall grade achieved, and are based on each individual learner's work. The continuous assessments provide for ongoing feedback to the learner and relates to the module curriculum.

| No. | Description   | MIMLOs | Weighting |
|-----|---|--------|-----------|
| 1   | Presentation<br>Each learner will be asked to prepare a presentation on a (different) topic related to recent technologies and trends in the field of cloud computing.                        | 3-7    | 20%       |
| 2   | Assignment<br>Learners are asked to develop a web application based on a real-world scenario. This must be deployed on a recommended cloud platform and configured as per the specifications. | 1-2, 8 | 30%       |
| 3   | Written exam that tests the theoretical aspects of the module.  | 1-8    | 50%       |

All repeat work is capped at 40%.

**Sample assessment materials**

Note: All assignment briefs are subject to change in order to maintain current content.

# Course Assignment

Worth - ??

Deadline: ??

## Description

Griffith College needs a **cloud-powered student management system** to store and manage all the necessary information about their students. The application should provide basic CRUD functionality to allow GCD staff to add, search, update, and delete student related information.

The college stores the following information for each student:

- Student number,
- Name (first name and last name),
- Address (temporary and permanent),
- Phone (home and mobile),
- Date-of-birth,
- Gender,
- Major (e.g., Computer Science),
- Course (e.g., HDWD),
- Study mode (part-time or full-time),
- Start date, and
- End date (expected end date of the course)

Above mentioned list indicates mandatory information which must be recorded for each student in the college. If you would like to store other information, feel free to add more fields.

- The system should present a welcome page which allows the user to login/logout with a staff account to use the student management system. You can allow your application to sign in users with Google Accounts.
- Once successfully logged in, the system should now welcome the user and present the CRUD options: **Add, Search, Update, Delete**.
- Upon clicking the **Add** link, a form should be presented to capture student's details. This will have an add button at the bottom. After entering student's details and clicking add button, the system should first validate the information and then store it into the Datastore. Upon success, a message should be displayed (e.g., New student added successfully!) with a link to home page. (Note: Duplicate student entries should not be allowed e.g., after validating the information you must check if the student already exists with the given student number).
- Upon clicking the **Search** link, the system will ask the user to provide student number to search. This will have a search button at the bottom. After entering student number and clicking search button, the system should display student information if found or display a message: "Student does not exist!", again with a link to return to the home page.



# Course Assignment

Worth – ??

Deadline: ??

## Marking Criteria

### Things to keep in mind:

- You must present a working application that covers the mandatory specifications listed above.
- All your **Java** and **JSP** code must be original and commented. You may be asked to explain any piece of code.
- In case, if you are re-using some standard code (e.g. searched online, book, etc.), remember to cite the source within the code using comments.

### Final application marks breakdown:

- 10% Visual Design – Navigation / Layout / HTML / CSS
- 20% Form (user input) validation
- 10% Add option
- 10% Search option
- 10% Update option
- 10% Delete option
- 20% User Authentication – Login/Logout functionality
- 10% Any extra feature added

### Deductions:

- 10% Project not submitted properly as per the instructions (see Deliverables section).
- 20% Per week late submission (maximum 2 weeks)

# Course Assignment

Worth – ??

Deadline: ??

- Upon clicking the **Update** option, the system will ask the user to provide student number for a student whose information needs to be updated. If student is found, existing information should be displayed in the form (like add student form) with an update button. Except the student number, user should be able to update all information. After updating and clicking the update button, an appropriate message should be displayed along with home link.
- Upon clicking the **Delete** option, the system will ask the user to provide student number for a student who should be deleted from the system. If student is found, the information must be listed as plain text or in a form and a delete button. Take appropriate steps afterwards.
- When sending information using forms, Add and Update forms must use the **POST** method while the Search option may use the **GET** method.
- When creating HTML forms, you must use appropriate input types such as *text* for name, *date* for date-of-birth, *radio* for gender etc.

## Deliverables

### What do you need to complete this assignment?

- A fully working installation of Google App Engine environment on your machine.
- **JSP** pages for receiving and presenting student information to the user.
- **CSS** to make your application look decent.
- **Servlets** to handle user requests. And
- **Datastore** to store student information.

### What/How do you submit?

- Deploy the final version of your application on Google App Engine.
- Take (small size) screenshots of your application (home page + one for each CRUD operation).
- Create a word file named **sms\_info.docx** which includes:
  - Overview: how the system is implemented (maximum one page),
  - The URL of your deployed application, and
  - Screenshots of the application.
- Create one zipped file that should only contain:
  - **sms\_info.docx** file (created as above) and
  - source code (include only **JSPs**, **CSS**, **Servlets**, and **web.xml**).
- For submission, name the zipped file as: **yourstudentnumber\_lastname.zip**
- Make the submission on moodle before the deadline.

**GRIFFITH COLLEGE DUBLIN**

**QUALITY AND QUALIFICATIONS IRELAND  
EXAMINATION**

**CLOUD SERVICES AND PLATFORMS**

**Lecturer(s):**

**External Examiners(s):**

**Date: XXXXXX**

**Time: XXXXXX**

**THIS PAPER CONSISTS OF FOUR QUESTIONS  
ALL QUESTIONS TO BE ATTEMPTED  
ALL QUESTIONS CARRY EQUAL MARKS**

### QUESTION 1

**Answer both (a) and (b)**

- (a) What are the three cloud computing models? Describe & give an example of each.

**(15 marks)**

- (b) What are the advantages & disadvantages of PaaS.

**(10 marks)**

**Total (25 marks)**

### QUESTION 2

**Answer all parts**

- (a) Name and explain Coffman's conditions for concurrent deadlock.

**(8 marks)**

- (b) Explain what a consensus protocol is. What are the challenges of reaching a consensus?

**(5 marks)**

- (c) Name and explain roles that a processor can play in Paxos algorithm.

**(10 marks)**

- (d) What does it mean if speed-up ratio  $S(N) > N$ ?

**(2 marks)**

**Total (25 marks)**

### QUESTION 3

**Answer (a), (b) and (c)**

- (a) What storage types are offered by Azure cloud? Explain the differences in use.

**(7 marks)**

- (b) Explain the journal storage model. With the aid of a diagram, explain how write operation works.

**(8 marks)**

- (c) GFS is unusual in that it replicates the locking mechanism. Justify this statement and explain measures that have been taken to safeguard the locking mechanism.

**(10 marks)**

**Total (25 marks)**

#### **QUESTION 4**

**Answer all parts**

- (a) Explain what the memcache is and how can it be used to increase the performance of web applications. Why consistency might be an issue when using memcache?

**(8 marks)**

- (b) Explain how automatic scaling of a cloud solution can be achieved with the use of listeners.

**(8 marks)**

- (c) Explain why is it impossible to have a system free of bottlenecks? Give an example of such situation.

**(5 marks)**

- (d) Explain the phases of the MapReduce programming technique.

**(4 marks)**

**Total (25 marks)**