# Solent University Unit Descriptor

## **Unit Code: COM528 Unit title: Object Oriented Design & Development**

### **Why is this unit important?**

The Object Orientated paradigm has deeply influenced commercial software development and new programmers need to be aware of the best practices for designing and documenting well structured, flexible, scalable and error free Object Orientated code. In addition, cloud computing is enabling Software as a Service (SaaS) products and programmers need to be able to deliver enterprise applications which are designed to be deployed as web services.

This unit will give you skills to hit the ground running when you enter the world of enterprise application development using object orientated technologies. In addition, it will introduced you to some of the most widely used enterprise java API's, version control, build and unit test frameworks. As a result of this unit you should be in a position to develop your own solutions, adopt or contribute to open source libraries and enhance your portfolio of experience for a future employer.

### **What you will learn on the unit**

This unit will build upon the object-oriented (OO) implementation skills developed at Level 4 with the aim of providing a sufficient understanding of OO development to undertake a small-scale OO software development project which interacts through web services with other SaaS components. You will also cover the requirements for good professional practice in OO design, development and testing will be emphasised. While core OO implementation techniques will be covered, the aim of this unit is to ensure you understand the software lifecycle from analysis and design through implementation to testing.

You will learn analysis using

· Use cases and tools bridging the gap between analysis and design; for example, using Unified Modelling language (UML) and robustness diagrams.

· Sequence diagrams.

You will learn Software design using

· Importance of low coupling and high cohesion in object-oriented software development.

· Object Orientated Design patterns such as Gang of Four patterns

· Model View Controller (MVC),

· Factory and Template patterns

· Inversion of control.

You will learn software development technologies and methodologies

· Version control (e.g. Git).

· Test driven development and build systems (e.g. Junit / Surefire / Maven).

· Container / Micro service / persistence technologies (e.g. JPA, Spring etc.)

· Web Service technologies (JAX-RS, JAXB etc.)

· Standard data storage classes e.g., array lists and maps. Data storage libraries e.g. the Java Collections Framework.

 Fundamental object-oriented development concepts, including polymorphism, abstract classes, and interfaces.

 Error handling with exceptions.

 Basic testing concepts: unit and integration testing, white-box and black-box testing, testing environments.

### **How you will learn**

IT lab-based sessions will deliver the content each week through discussions and exercises.

Each week we will work on incrementally developing a design for a complete enterprise system. Enterprise systems are often developed in layers with each layer being independently developed and tested before integration into the full system.

You will keep your work in your own git hub repository and incrementally add to this software as your design is developed.

In each class we will have a brief lecture and demonstration followed by a set of exercises to help you apply the technology or technique which has been introduced.

Once we have learnt a technology, we will begin to use it to develop a layer of our proposed system. Ultimately, we will have a fully working enterprise App.

You will also be asked to work in teams to develop complimentary client and server technologies which demonstrate how SaaS components are developed in practice.

It is expected that you will review the material and work on the tasks in your own time prior to the next session. This independent learning will strengthen your skills and is essential if you want to fully embrace the subject area. It is also anticipated that you will read around the subject area. Guidance will be given through the VLE Solent Online Learning to help with this.

Reviewing your work each week will provide formative feedback on your progress.

### **How much time the unit requires**

This unit is a 20 credit unit and students are expected to study for 200 hours (which equates to 10 hours per credit).This total learning time is made up of contact time, directed learning tasks, independent learning and assessment activity.

### **How you will be assessed**

#### **Tasks which help you to learn and prepares you for summative tasks (Formative):**

Formative feedback will be given continuously on the laboratory exercises, and peer review and discussion will also assist. You will have the opportunity to obtain feedback on the assignment prior to submission.

#### **Tasks which count towards your degree (Summative):**

There will be two assessments: a practical mini project, working in teams and an individual main project, following on from the first, which is supported by documentation.

The first assignment will allow you, working as a member of a development team, to incrementally produce a small piece of software following a development approach from analysis of a specification through design to implementation and testing. Formative feedback will be given continuously on the laboratory exercises, and peer review and discussion will also assist. You will have the opportunity to obtain feedback on the assignment prior to submission. The assessment is anticipated to focus on understanding of the development lifecycle rather than complexity in implementation.  You will present this in your team to the tutor and cohort.

Using the object-oriented implementation concepts learned in the unit, you will individually extend the first project to a given set of requirements. The report should discuss your design and execution and is supported by your software model. You will be expected to evaluate and reflect on the development process.

#### **When assessment does not go to plan** Re-assessment task must demonstrate achievement of the learning outcomes covered by the original assessment task. Therefore, you will re-submit your reworked assessment subject to guidance and feedback from the tutor.

### **What you will be able to do after the unit**

1. Identify appropriate analysis, design, implementation and testing techniques in object-oriented software development.
2. Analyse problems in object-oriented software design and development with solutions.
3. Evaluate the process of developing a piece of object-oriented software.
4. Systematically apply and document the complete development lifecycle (from analysis through design, implementation and testing) of an object-oriented application.
5. Produce documentation according to standards expected in current professional situations.
6. Collaborate with others on a mini project.

**How this relates to the dimensions of Solent’s Real-world curriculum framework**

|  |  |  |
| --- | --- | --- |
| Dimensions | How Students Learn | How students are assessed |
| Students are challenged to think in critical, creative and applied ways | Students work together to solve problems | Students collaborate on a mini project |
| Students are inspired to do research through inquiry, curiosity and problem-solving | Students analyse problems | Assessment demonstrates understanding |
| Students experience an intellectually stimulating curriculum which inspires them to learn for life | Students analyse problems | Assessment demonstrates understanding |
| Students learn from authentic, engaging and programmatic assessment | Students learn about the requirements of professional standards | Their understanding of professional standards is assessed. |

### Summative assessment details

|  |  |  |
| --- | --- | --- |
| AE1 | Weighting: | 40% |
|  | Assessment type: | Group Mini project presentation |
|  | Aggregation: | Aggregated to AE2 |
|  | Length/duration: | Software model, with comments. 5 minutes each |
|  | Online submission: | Yes |
|  | Grade marking: | Yes |
|  | Anonymous marking: | No |

|  |  |  |
| --- | --- | --- |
| AE2 | Weighting: | 60% |
|  | Assessment type: | Individual Software Project with supporting documentation |
|  | Aggregation: | Aggregated to AE1 |
|  | Length/duration: | 2000 words |
|  | Online submission: | Yes |
|  | Grade marking: | Yes |
|  | Anonymous marking: | No |

### Unit Author: Dr Craig Gallen

|  |  |  |  |
| --- | --- | --- | --- |
| Unit Title: Object Oriented Design & Development | | | |
| Credit Points: | 20 | Unit Code: | COM528 |
| FHEQ Level: | 5 | School/Service | SMAT |
| Unit Delivery Model: | CD | Max/Min student numbers | N/A |
| Unit Leader: | Dr Craig Gallen | | |
| HECOS code | 100374 | | |

### Unit change history:

|  |  |  |  |
| --- | --- | --- | --- |
| Unit Approved/Year Implemented/Code | July 2019 | 2020/21 | COM528 |
| Unit modified/Year Implemented/Code |  |  |  |
| Add extra rows as required |  |  |  |