

**UNIVERSITY OF BOLTON**  
**CREATIVE TECHNOLOGIES**  
**BSC (HONS) GAMES PROGRAMMING**  
**SEMESTER ONE EXAMINATION 2018/2019**  
**SOFTWARE ENGINEERING**  
**MODULE NO: GAP5002**

Date: Wednesday 16<sup>th</sup> January 2019

Time: 14:00 – 16:00

---

**INSTRUCTIONS TO CANDIDATES:**

There are **FIVE** questions on this examination.

Answer **THREE** questions. All questions are worth 25 marks. During assessment your marks out of 75 will be converted to a percentage.

Calculators may be used for this examination.

Note: Where code is requested, minor syntax errors will **not** be penalised.

---

Creative Technologies  
BSc (Hons) Games Programming  
Semester One Examination 2018/2019  
Software Engineering  
Module No: GAP5002

**Question 1**

Software engineering encourages a number of design principles to aid software development. With the aid of suitable examples, outline the following design principles:

- a) Single Responsibility Principle (SRP). **[5 marks]**
- b) Open Closed Principle (OCP). **[5 marks]**
- c) Liskov Substitution Principle (LSP). **[5 marks]**
- d) Interface Segregation Principle (ISP). **[5 marks]**
- e) Dependency Inversion Principle (DIP). **[5 marks]**

**PLEASE TURN THE PAGE....**

Creative Technologies  
BSc (Hons) Games Programming  
Semester One Examination 2018/2019  
Software Engineering  
Module No: GAP5002

**Question 2**

A player in an adventure game can carry and use a number of artefacts. A player can carry and use one artefact at a time, but a player can change artefacts at any time during the game.

- a) Based on the *Strategy* pattern, draw a class diagram that relates a *Player* class with an *Artefact* strategy class, with artefacts: *Torch* and a *Matches*. The *Player*'s constructor should allow an *Artefact* strategy pointer to be stored. Ensure there is a consistent function name used for using artefacts. **[6 marks]**
- b) Based on your class diagram, write C++ code for a *Player*'s constructor, using an inline function definition. **[3 marks]**
- c) Write the code that would instantiate a *Player* object with a *Torch* and then change the *Player*'s artefact to *Matches*. **[6 marks]**
- d) Write down the additional code for releasing allocated memory, when artefacts are changed. Indicate in what function, and where, such code should be placed. **[4 marks]**

In this scenario, it is likely that a pointer to a *Game* object is also required. As such, a *Game* pointer may be included, as a parameter, when *using an artefact*.

- e) Draw outline classes, i.e. without attribute / function details, to highlight the 'new' design. **[3 marks]**
- f) Explain why including an additional pointer to *Game* impairs the overall design. **[3 marks]**

Please note that pre-processor statements are **not** required, for any code.

**PLEASE TURN THE PAGE...**

Creative Technologies  
 BSc (Hons) Games Programming  
 Semester One Examination 2018/2019  
 Software Engineering  
 Module No: GAP5002

### Question 3

A program that stores Items in an Inventory has the following code fragment:

```
class ItemCreator{
    Drink *drink;
    Food *food;
    Torch *torch;
public:
    void makeAnItem(int type){
        switch(type){
            case 0:
                drink = new Drink();
                break;
            case 1:
                food = new Food();
                break;
            case 2:
                torch = new Torch();
                break;
            default:
            }
        }
        //more code, followed by more functions
    };
};
```

- a) With regard to the above code, outline its major limitation, from a design point of view. **[3 marks]**
- b) It has been suggested that the above code could be improved by using the *Factory* pattern. Draw a class diagram, using the *Factory* pattern, which includes an *ItemFactory* class and an *ItemFactoryImplementation* class, which enables Items to be created using an abstract Item class. **[10 marks]**
- c) The items created from the Factory in b) would need to be stored in an inventory. If the number of items in the inventory can vary greatly, suggest an appropriate data structure, and justify its use, that could be used to store such items. **[4 marks]**
- d) The items inherit from a Sprite class too. If a large number of items, with the same Sprite image, are to be created, using an outline class diagram, i.e. without attributes / functions, outline a pattern that could help reduce storage space, and suggest a suitable data structure, with fast search capabilities, that could be used to achieve this. **[8 marks]**

Please note that pre-processor statements are **not** required, for any code.

**PLEASE TURN THE PAGE....**

Creative Technologies  
BSc (Hons) Games Programming  
Semester One Examination 2018/2019  
Software Engineering  
Module No: GAP5002

**Question 4**

a) In terms of software development, explain what Agile development is, and outline three advantages compared to Waterfall development. **[9 marks]**

b) Outline what is meant by the terms:

- White box testing.
- Black box testing.
- Unit testing

**[6 marks]**

Advocates of Extreme Programming, (XP), emphasise the need for automated testing.

c) With the aid of example C++ *test* code, detail how automated unit testing can be included within a project, and outline when such an approach is particularly useful. **[10 marks]**

**Question 5**

An accountant, placed in charge of a group of programmers, states: *“A good way of measuring programmers’ productivity is to obtain the number of lines of code produced, per programmer, per week.”*

Discuss this statement, particularly in relation to the potential of frequent changes to a code base, when prototyping. **[25 marks]**

**END OF EXAMINATION**