

## Semester One of Academic Year (2015---2016) of BDIC

### 《Software System Architecture》

Module Code: COMP3029J

### Exam Paper A

Exam Instructions: Answer ALL Questions

**Honesty Pledge:**

I have read and clearly understand the Examination Rules of Beijing University of Technology and University College Dublin and am aware of the Punishment for Violating the Rules of Beijing University of Technology and University College Dublin. I hereby promise to abide by the relevant rules and regulations by not giving or receiving any help during the exam. If caught violating the rules, I would accept the punishment thereof.

Pledger: \_\_\_\_\_

Class No: \_\_\_\_\_

BJUT Student ID: \_\_\_\_\_

UCD Student ID \_\_\_\_\_

.....

**Notes:** The exam sheet covers 5 type of questions, and 18 pages, totally 100 points. During the exam, please use the attached answer sheet and scratch sheet to answer the questions

Exam Sheet Score Collecting Table (Teacher only)

No.	一	二	三	四	五	六	七	八	Total Score
Full Score	30	10	20	20	20				
Score									

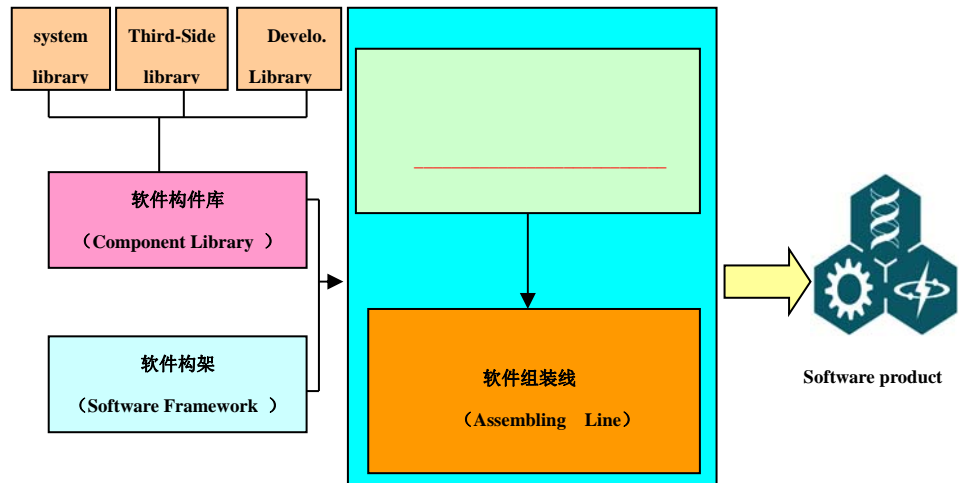
(Score)

一、 Fill in the Blank (30 Points, 1 Point Per Each Blank)

1. Designed and developed by the professional engineering specialists, a software system/application is referred as to a product that delivers computing device potential, and is the logical & structural organization of \_\_\_\_\_;
2. Software Architecture – is a meaningful representation of design concept and a modeling blueprint for the software product that is to be built, is a skeleton plan of \_\_\_\_\_ and the decision of \_\_\_\_\_, embodied in its components(building blocks, parts, and structural elements):
  - A decision on the division of \_\_\_\_\_;
  - A design specification to guide and aid in the following step of software procedure;
3. Issues faced by software building: 1) a software is complex; 2) a software is uncertainly developed; 3) a software is long life-cycle, high-cost, poor-quality; Software architecture design is one of the most fundamental techniques and the most important procedure, bridging the gap between software’s \_\_\_\_\_ and system implementation (coding).
4. Software Product/System’s production-line manufacturing: adapts a production-line of software products, i.e. by means of \_\_\_\_\_, standardized specification and behavior code to develop and implement the software’s\_\_\_\_\_ that exposes to all kinds of application scenario, such a pattern is used to deploy the components into the open framework, to composite into the expected products
5. The design methodology of software architecture evolves from the following stages:
  - Functionality program design;
  - Structure program design;
  - \_\_\_\_\_
  - \_\_\_\_\_

-- SOA program design.

6. Modern software production is depicted as the following diagram, fill the blank in the name of related item \_\_\_\_\_ in the textbox.



7. The connecting topologies among the components are depicted as the below diagram, in figure A, the connecting mechanism is called as software architecture with \_\_\_\_\_ connecting while in figure B, the connecting mechanism is named as software architecture with \_\_\_\_\_ connecting.

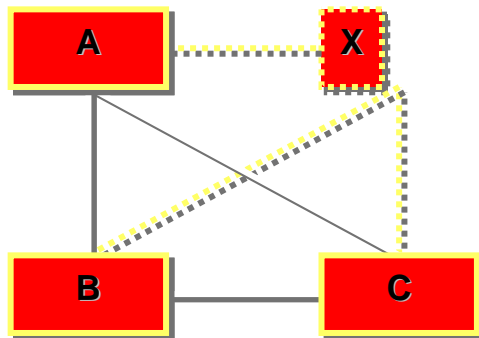


Figure A

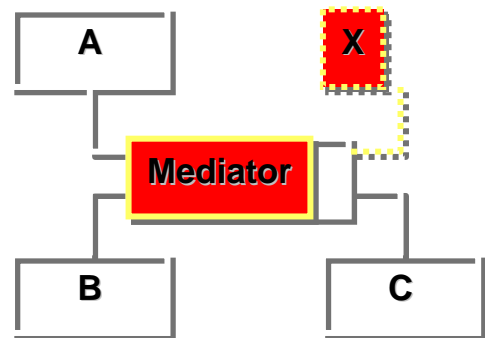


Figure B

8. Software Design Style – reflects the connecting and organizing mechanism of components, different kinds of software style includes \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_.
9. Evaluation & Analysis of Software Architecture – review on design documents,

design procedure, code, implementation procedure, to detect \_\_\_\_\_ of software, promote the quality of software, and \_\_\_\_\_ ahead of the disaster caused by error of design, and choose the better one or \_\_\_\_\_ design.

10. Quality attributes of a software system covers multi-categories, ranging from usability, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, testability, ease of creation, transplantability, reusability, integrability)
11. Graphic representation is a simple and a widespread of tool for software architecture representation, within this scope, Rectangle items (with arrow, without arrow, with solid-arrow, with dash-arrow, ....) is used as the model notation of component and \_\_\_\_\_.
12. Graphic User Interface (GUIs): is an interactive interface between the user and the computer models, GUIs offer the user user-friendly, ease-use, helpful media to the end users: Generally speaking, GUIs are characterized by the \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_ items etc..

Score

**二、 True/False Choice** (10 Points, 1 Point Per Question, If Choice of True, Fill ✓ in the Blank; If Choice of False, Fill X in the Blank)

1. Software architecture is not only the representation and modeling description of software system/application, but also the runnable software application( );
2. Within the scope of structural software architecture design, the entry point of principle component is a main() function ( );
3. N-Tiers software architecture style is referred to as a structure of software system components, each layer will offer interactive service for the above layer,

- and non-neighboring layers will not interoperate ( );
4. In software architecture, middleware component will not be allowed ( ).
  5. The style and patter of a software architecture can be applied to the scenario of design for other software systems ( );
  6. In the field of designing software architecture with object-oriented analysis and design method, the fundamental units of a system is the functions ( );
  7. The method of representation and description of software architecture includes formal and informal approaches: UML is an informal method while ADL is a formal one ( ) ;
  8. With the design method and approach of distributed software architecture, it’s possible to implement inter-connecting, interactive and interoperating among the components of heterogeneous platform ( ) ;
  9. The evaluation and analysis methods of software architecture include SAAM, ATAM and ARID etc.. ( );
  - 10 In MVC patterns, there are presentation layer, model, and controller modular( );

Score

### 三、 Question and Answer ( 20 Pints, 5 Points Per Question)

- 1 (5 points) What’s software architecture? Describe the principle content and fundamental elements.
- 2 (5 points) What’s the distributed software application? Describe the benefits and shortages of distributed software architecture patterns.
- 3 (5 points) Describe the software architecture connected by the middleware container with the demo diagram.
- 4 (5 points) Describe the roles and tasks of a component in software architecture.

Score

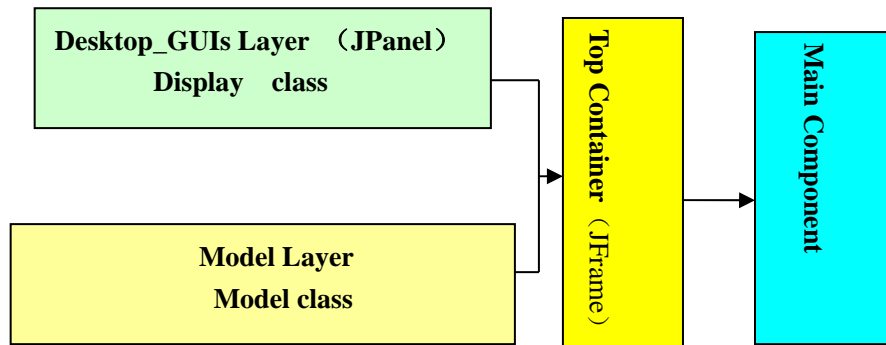
#### 四、Programming of Design and Coding Comprehension (20 Points, 5 Points Per Question)

1 (5 points) Design and implement the software architecture of a software application with 4-tier pattern, including client-side layer, presentation layer, business logic layer, and data resource management layer by using MFC VC++/Java/C#/C programming language. You are request to describe the components of each layer and call procedure.

2 (5 points) A CRC (Component Responsibility Card) is depicted as follow, you are required to implement the specification of the card by using C/ C++/C# or Java programming language.

<b>component name: Rocket</b>	
<b>super classes:</b>	
<b>subclasses:</b>	
<b>Responsibilities</b> 1 Take charge of describing basic attributes, InitInstance, Launch, Fly, ExitInstance,... of A Rocket Model	<b>Collaborators</b> class Rocket_Display
<b>Attributes</b> int H, V int x, y, z,	class Rocket_Display { Rocket_Display() { } .... } .....
<b>Responsibilities</b>	
C Rocket(CRocket_Display rocket); // Constructor void DrawRocket ( ) ;               // Drawing Rock void InitInstance();               // Initial handling void ExitInstance ();               // exiting handling void Launcher();               //Launching handling void Fly();               // Flying handling .....	

3 (5 points) The software architecture of an application system is depicted as follows, write the results of running the application with Java programming language.



```

class Exam_App {

    public static void main() {

        new Exam_Frame();

    }

}

class Exam_Frame extends JFrame {

    Container contentPane;

    Display display;

    Model model;

    Exam_Frame() {

        contentPane = this.getContentPane();

        display = new Display();

        model = new Model(display);

        setSize(1024,768);

        setVisible(true);

    }

}

class Display {

    BufferedImage buffer = null;

    ...

    Display()

    {

```

```
        buffer = new BufferedImage(1024,768, BufferedImage.TRANSLUCENT);
    }

    public void paintComponent(Graphics g)
    {
        g.drawImage(buffer,0,0, null);
    }

    ...
}

class Model
{
    BufferedImage image;

    Graphics g;

    Display panel = null;

    Model( Display pane)
    {
        this.panel = pane;

        g = panel.buffer.getGraphics();

        g.drawString( "I am participating the test of Software Architecture ", 10, 10);

        g.drawString( "I will Success!!", 10, 50);

    }

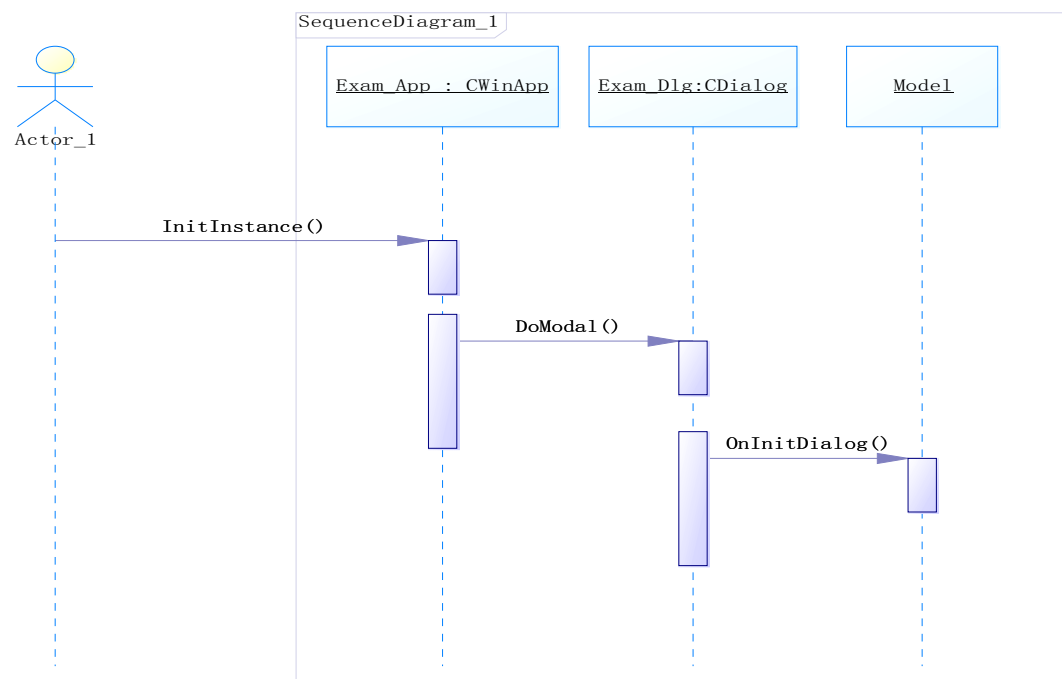
    ....
}
```

**The results are:**

4 ( 5 points ) The software architecture (time-sequence diagram) of a VC++ MFC application program is described as follows, please complete the code in the blank according to the requirement. Exam\_App component as a main component is used to assembly the GUIs component: Exam\_Dlg; GUIs component as a visual user interface is used to call and capture the data from the Model component: Model, via a function:



Model\_Display) .



```

#include<afxwin.h>

class Exam_App : CWinApp
{
public:
    virtual BOOL InitInstance();
};

Exam_App app;

class Exam_Dlg: CDialog
{
Public:
    Model *model;

public:
    void OnInitDialog();
    void Model_Display();
}
    
```

```
};  
  
class Model  
{  
    public:  
        Model();  
        String model_data();  
};
```

Giving and coding the function: Model\_Display in class Exam\_Dlg, in which Model\_Display call and return data from model\_data() in class Model.

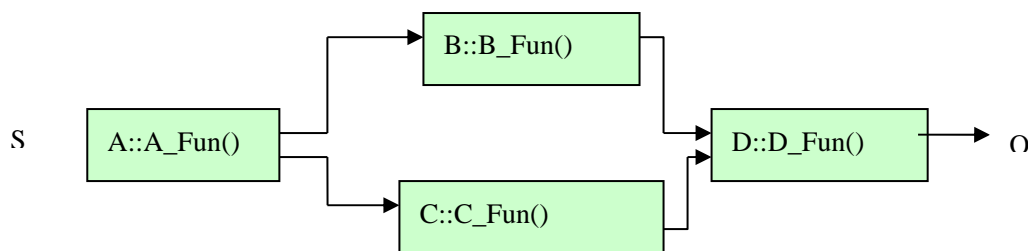
```
void Exam_Dlg::OnInitDialog()  
{  
    // Giving and coding the pointer to the class Model;  
  
}  
  
void Exam_Dlg::Model_Display()  
{  
    // Giving and coding the procedure of calling and returning the data from String  
        model_data() in the class Model.  
  
}
```

Score

## 五、Comprehensive Designing & Programming Application

(20 Points, 10 Points Per Question)

1 (10 points) The software architecture of pipeline-filter style application is depicted as the diagram, A::A\_Fun, B::B\_Fun, C::C\_Fun, D::D\_Fun represent the function of A, B, C, D (input / output is string type, S as an original input String, and O as a final output result String).



- (1) Coding the components A, B, C, and D by C, or VC++ MFC, or Java programming language
- (2) Design the software architecture, including a main component to assemble the component A, B, C, D )

2 (10 points) E-Schoolbag is a Web-based distributed application system, including client-side app, server-side component and data resource administration-side component, in which the users acquire the e-textbook of Word or PDF type. Design the software architecture such as follows:

- (1) Design the software architecture, including client-end layer, Web service layer, data resource layer, by using graphic representation methods)
- (2) Describe Technical Reference Model and Architecture, you are required to write down the planning and configuration of hardware environment, including client-side hardware environment, server-side environment and database system environment.
- (3) Coding the components of each layer, at least, write down the function of each component)