**TCS2411/TSE2101 Software Engineering Fundamentals**

**Tutorial 6**

**Part A: Discussion**

Topic (Lecture 6): Design Concepts, Data Modeling and Interface Design

* Do you design software when you “write” a program? What makes software design different from coding?
  + Coding is the technical aspect of the software creation process that occurs after the design is completed. Design is the process by which we decide the functionality and how a program will work to achieve the desired goal.
* How do we assess the quality of a software design?
  + The quality of software is assessed based on design even before it is implemented.During design, the quality is assessed by conducting a series of technical reviews.
* Describe how the CRC cards can be used to identify classes in requirements analysis.
  + CRC include 3 part: class, responsibilites and colaborators,
  + The CRC process is based on going through use cases (or specific scenarios built from use cases), and using these to discover objects, responsibilities, and collaborators.
  + It can identify classes during the process, main classes is identified first based on the use cases folow by responsibilities and colaborators
  + Sometimes, we'll also identify a collaboration with a new object type that doesn't have a card yet -- this helps discover new classes
* Describe the following types of cohesion:
* Functional
  + This is achieved when all the code that computes a particular result is kept together - and everything else is kept out
* Layer
  + All the facilities for providing or accessing a set of related services are kept together, and everything else is kept out
* Communicational
  + All the modules that access or manipulate certain data are kept together (e.g. in the same class) - and everything else is kept out

5. Match the following coupling categories with its descriptions. You should select the best answer from Table 1.

Table 1 – List of Coupling Categories

|  |  |  |
| --- | --- | --- |
| Common coupling | Data coupling | Control coupling |
| Routine call coupling | Type use coupling | Couple coupling |
| Inclusion/Import coupling | Content coupling | External coupling |
| Stamp coupling | Unusual coupling | Good coupling |

**Solution:**

5a) \_\_\_Content coupling\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Occurs when one component “surreptitiously modifies data that is internal to another component
* This violates information hiding – a basic design concept.

5b) \_Common coupling\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Occurs when a number of components all the make use of a global variable.
* Although this is sometimes necessary (e.g. for establishing default values that are applicable throughout an application), common coupling can lead to uncontrolled error propagation and unforeseen side effects when changes are made.

5c) \_\_Control coupling\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Occurs when operation A () invokes operation B() and passes a control flag to B.
* The control flag the “directs” logical flow within B.
* The problem with this form of coupling is that an unrelated change in B can result in the necessity to change the meaning of the control flag that A passes.
* If this overlooked, an error will result.

5d) \_\_\_Stamp coupling\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Occurs when **ClassB** is declared as a type of an argument of an operation of **ClassA**.
* Because **ClassB** is now a part of the definition of **ClassA**, modifying the system becomes more complex.

5e) \_\_\_Data coupling\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Occurs when operations pass long strings of data arguments.
* The “bandwidth” of communication within classes and components grows and the complexity of the interfaces increases.
* Testing and maintenance are more difficult.

5f) \_\_\_\_Routine call coupling\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Occurs when one operation invokes another.
* This level of coupling is common and is often necessary.
* However, it does increase the connectedness of a system.

5g) \_\_\_Type use coupling\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Occurs when component A uses a data type defined in component B.
* For example: this occurs whenever “a class declares an instance variable or a local variable as having another class for its type”
* If this type definition changes, every component that uses the definition must also change.

5h) \_\_\_Inclusion/Import coupling\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Occurs when component A imports or includes a package or the content of component B

5i) \_\_\_\_External coupling\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Occurs when a component communicates or collaborates with infrastructure components (e.g.: operating systems functions, database capability, and telecommunications functions).
* Although this type of coupling is necessary, it should limit to a small number of components or classes within a system.

1. Why is it important that we understand the skills of the users when creating the user interface of a software?

* To make sure user can communicate well with the user interface.
* To make the user interface is user friendly to them.

**Part B: Modeling**

Topic (Lecture 6): CRC cards, Class Diagrams and Sequence Diagrams

Go to these links and learn more on how to draw class diagram, sequence diagram and use CRC cards.

<https://www.youtube.com/watch?v=Bxgn6qJ-bYY>

Using CRC cards

<https://www.youtube.com/watch?v=5Q0ZM6ODBHw>

UML Class Diagrams

<https://www.youtube.com/watch?v=18_kVlQMavE>

5 Steps to Draw a Sequence Diagram

<https://www.youtube.com/watch?v=4WDbte6cPa8>

UML Sequence Diagram

<https://www.youtube.com/watch?v=XIQKt5Bs7II>

UML Behavioral Diagrams: Sequence

Form into your assignment group and create class and sequence diagrams for your project.