Name …………………………………..

ID ……………………

Group ………………..

Mark ……… / 60

TA ………………..

Cairo University

Faculty of Computers and Information

Midterm Exam

No Extra Answer Papers

Course Name: CS352 - Software Engineering II **Date:** **22 May 2015**

Total Marks: 10 marks – 60 points **Duration:** **1 hour**

Question-1 General – Choose the MOST GENERAL answer. (20 points, 2.5/question)

1. Waterfall model is suitable for applications:

   (a) II and IV (b) II, III and IV **(c) II, III** (d) I, IV (e) I, III

   I. Whose requirements change frequently

   II. That have stable requirements

   III. That are well understood and company have previous experience with similar ones

   IV. That are exploratory and require a lot of experimenting and trial and error

1	2	3	4	5	6	7	8
c	e	b	d	a	c	c	d

2. Linux architecture follows the ………. Style

   (a) Pipe-and-filter (b) Peer-2-peer (c) Event-based (d) Repository **(e) Layered**

3. Whose job is to set the technical vision for the project, mentor the technical staff on design issues, and monitor design and implementation artifacts?

   (a) Project Manager **(b) Software Architect** (c) QA Engineer (d) Scrum Master

4. Which design pattern would you use for the following case: You want to make it easy to define new operations (algorithms) that work on a collection of objects of varying classes that inherit from a common parent. And these operations have to be performed on all the collection elements without knowing the type but they treat different kinds of elements differently?

   (a) Factory (b) Bridge (c) Template Method **(d) Visitor** (e) Strategy

5. A compiler system is an example of the following software architecture style:

   (a) Pipe-and-filter (b) Layered (c) Even-based (d) Repository (e) MVC

6. Scrum retrospective meeting is done to:

   (a) Allow every team member to say what he did yesterday, what he will do today & any problems he has

   (b) Plan the next sprint and select the user stories to implement next

   (c) Discuss how the sprint went and decide what and how the team will improve in the next sprint.

   (d) Discuss with the customer the most important user stories to pick for next sprint.

7. An organizational process focus, a company-wide process program, a repository of process assets and infrastructure and training on process model are key performance areas (KPAs) to get CMMI certification at level:

   (a) Level 1 (b) Level 2 **(c) Level 3** (d) Level 4 (e) Level 5.

8. Which design pattern can best implement this requirement? We would implement a simple news board with hot topics. A users can register to a topic he likes. Whenever any new news are posted on the topic, all the registered users will be notified and they can acquire the updated news.

   (a) Factory (b) Visitor (c) Adapter **(d) Observer** (e) Strategy

Question-2 Blochchain and REST Web Services (10 points)

1- BREIFLY (1) **Explain** what blockchain technology is, (2) **How it** works, (3) **How** it achieves trust and prevents data alteration and (4) **One suitable** application that can benefit from blockchain and (5) **Why** this application benefits from blockchain. 5 points

There can many ways to answer, especially parts (4) and (5). 1 point / item

We explained this in one lecture. The required answers are brief ones that show s/he understands the idea.

(1) Blockchain is a peer to peer distributed ledger technology that provides an efficient, cost-effective, reliable, auditable and secure system for conducting and recording transactions. It is shared among all involved parties. Each party (node) keeps a copy of the entire blockchain.

(2) How it works. (1) Someone requests a transaction (2) The request is broadcasted to the nodes members of the P-2-P network holding the blockchain (3) The nodes validate the requester's status and the transaction (4) When the transaction is verified, a block is created for it and is added to the blockchain and linked to the previous one.

(3) Trust is achieved by (1) linking each block to the previous one by a hash code for the block content that includes the hash of the previous one. (2) To alter a block or change it, you need to change the entire blockchain (3) But even so, the majority of the nodes will reject it as they have the correct copy (4) So you must alter the blockchain in more than 50% of the nodes. (5) This is computationally impossible.

(4) One application. Blockchain technology can be used to store reliably health records and share them with healthcare service providers.

(5) Why? Healthcare records are immutable and should not be altered with. The entire history of the patient and be securely stored in a blockchain shared among providers of healthcare for him or her.

2- (1) **Explain two** of the benefits of web services and (2) **Explain three** of the features of REST services. 5 points

***Benefits: There could be many correct answer. Two only are required. Most common ones are:**

(1 point per benefit – SOME Explanation must be provided)

1) **Interoperability.** Web services allow various applications to talk to each other and share data and services even if made with different technologies. For example, a C# or .NET application can talk to Java web services and vice versa. Web services make the application platform and technology independent

2) **Standardization.** Web services use standardized industry standard protocol for the communication like HTTP and XML. This standardization gives has many advantages like using common tools, having a wide range of choices and using existing low-cost Internet for implementing web services instead of propeitry

3) **Composition & Integration.** An application can be built by composing different services together and integrating them in a way that serves its model or logic. Minimal coding will be required to deploy these readymade services.

- **Reusability.** Web services provide standalone self-contained applications that can be employed and called from within other services using standard Internet protocols. It is also possible to deploy or wrap legacy systems as services accessible to modern Java based or JS based apps.

Features of REST services: There could be many correct answers. Three only are needed.

(1 point per feature)

1) **Client-server.** REST services employ client-server model. A server component offers a set of services, listens for requests upon those services. A client component sends a request for a service to the server via a

connector (HTTP usually). The server either rejects or performs the request and sends a response back to the client.

2) **Stateless.** In REST services, server has no client state. Any session state is held on the client. Each request is independent of previous ones and has enough information to process it (self-descriptive).

3) **Cacheable.** The data within a response to a client's request are labeled as cacheable or non-cacheable. If a response is cacheable, then a client cache is given the right to reuse that response data for later, equivalent requests.

4) **Unique Identifier.** REST services represent application functionalities, which are treated as resource. Each resource has a unique URI identifier similar to web pages.

Question-3 Design Principles and Deign Patterns (30 marks)

Solve ONLY two of these questions

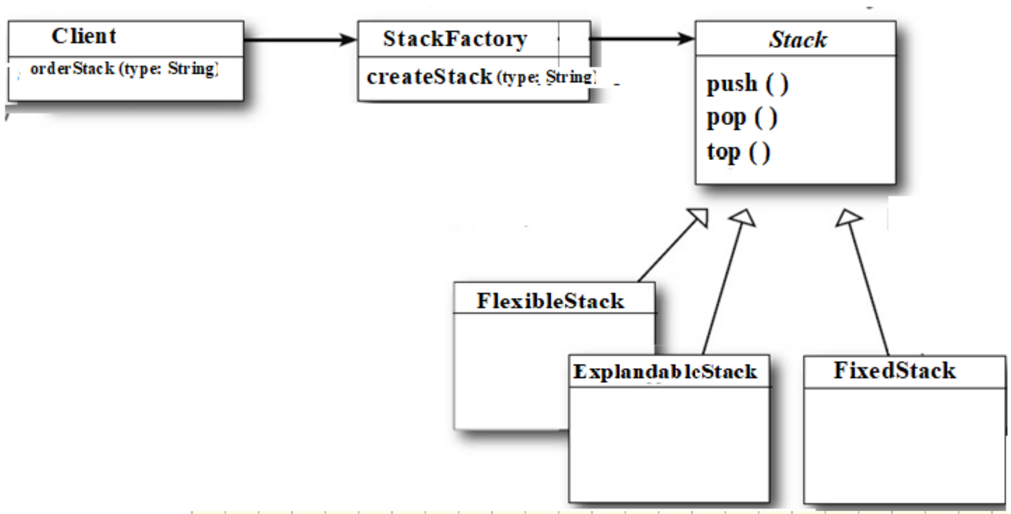
1- **Apply** the *Factory* pattern to solve the following problem: An application needs a stack of integers in performing its functions. There are three types of stacks. There is (1) a FIXED-size stack that holds up to a maximum number of elements, (2) an EXPANDABLE stack that always increases its size when it is full but does not shrink and (3) a FLEXIBLE stack that expands and shrinks according to the size of data stored in it. Different versions of the application use different types of stacks. But the rest of application functions are not aware of what type of stack they are dealing with. 15 points

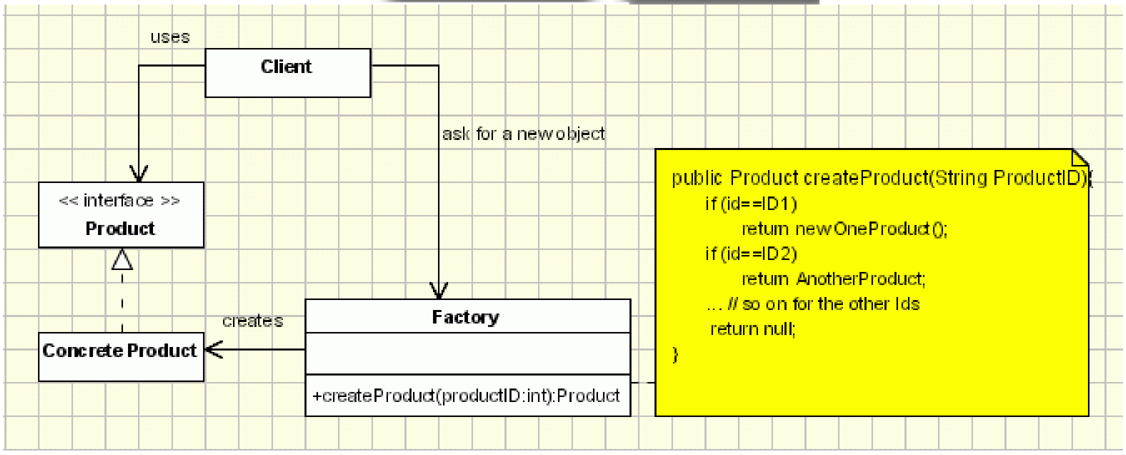
(a) **Explain** how *Factory* pattern can be used to create stacks. **(8 marks)**

The application will be associated with a Stack Factory that is responsible of creating and returning a new stack upon request. This will separate Stack creation from Stack consumption. Upon request with a passed type, Stack Factory will create the required Stack type and return it. The rest of client code will not know which kind of stack it is. Even at run time, the type of needed Stack can be defined. Design will be open for extension but closed to modification. So, if we want to add a new Stack kind, nothing changes in the client. But the Stack Factory changes to be able to create the new type.

- The client needs a product, but instead of creating it directly using the new operator, it asks the factory object for a new product, providing the information about the type of object it needs.
- The factory instantiates a new concrete product and then returns to the client the newly created product(casted to abstract product class).
- The client uses the products as abstract products without being aware about their concrete implementation.

(b) **Draw** a class diagram of the solution, showing all the methods necessary to the pattern and their signature (name, parameters and return type). **(7 marks)** – Note, StackFactory could be connected to children Stacks instead.





```
public Product createProduct(String ProductID){
    if (id==ID1)
        return new OneProduct();
    if (id==ID2)
        return AnotherProduct;
    ... // so on for the other Ids
    return null;
}
```

2- Given the following incomplete implementation of a program that implements the Template Method design pattern, **complete this code** by writing the missing parts. And **draw a UML class diagram** for the pattern showing all the details of classes, methods and their features. 15 points

Assume that methods `printTicketDetails` and `printHotelDetails` print only simple messages.

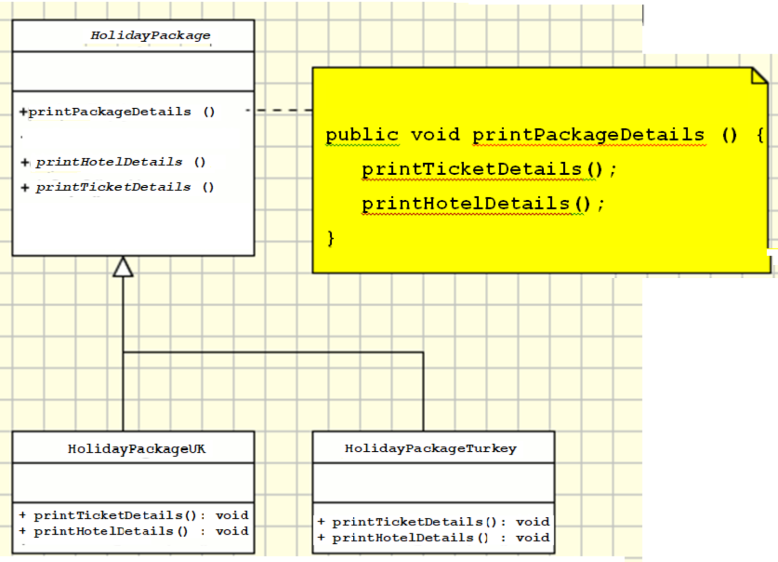
```
public abstract class HolidayPackage {
   private int price;
   public void printPackageDetails () {
      + printTicketDetails(): void
      + printHotelDetails()  : void
   }
   public abstract void printTicketDetails ();
   public abstract void printHotelDetails ();
}
```

Solution: 2.5 x 4 missing methods + 5 for UML

```
public class HolidayPackageTurkey extends HolidayPackage {

   ..................................... // (1) Add code here
  public  void printTicketDetails (){           // 2.5 marks
     System.out.println ("Turkey Ticekt");
  }
  public  void printHotelDetails () {           // 2.5 marks
     System.out.println ("Turkey Hotel");
  }
}
```

```
public class HolidayPackageUK extends HolidayPackage {

   ..................................... // (2) Add code here
  public  void printTicketDetails (){           // 2.5 marks
     System.out.println ("UK Ticekt");
  }
  public  void printHotelDetails () {           // 2.5 marks
     System.out.println ("UK Hotel");
  }
}
```

5 marks for correct class diagram showing abstract methods and inheritance and concrete methods in children



```
public void printPackageDetails () {
   printTicketDetails();
   printHotelDetails();
}
```

3- (a) **Which design principle** is violated in the code below? And **why?** 15 points

Interface Segregation Principle. (2)

Because Interface is polluted with functions not needed for a normal Xerox machine. (2)

(b) **Modify the code** to properly apply the violated principle.

```
public interface IPhotoCopier {
   public void print();
   public void scan();
   public void photoCopy();
}

public class XeroxNormalMachine
   implements IPhotoCopier {
     public void print() {}
      public void scan()  {}
        public void photoCopy() {
          System.out.println("copy");
     }
}
```

```
public class XeroxComboMachine
  implements IPhotoCopier {
    public void print() {
      System.out.println("Combo Print");
    }
    public void scan() {
          System.out.println("Combo Scan");
     }
     public void photoCopy() {
          System.out.println("Combo copy");
     }
 }
```

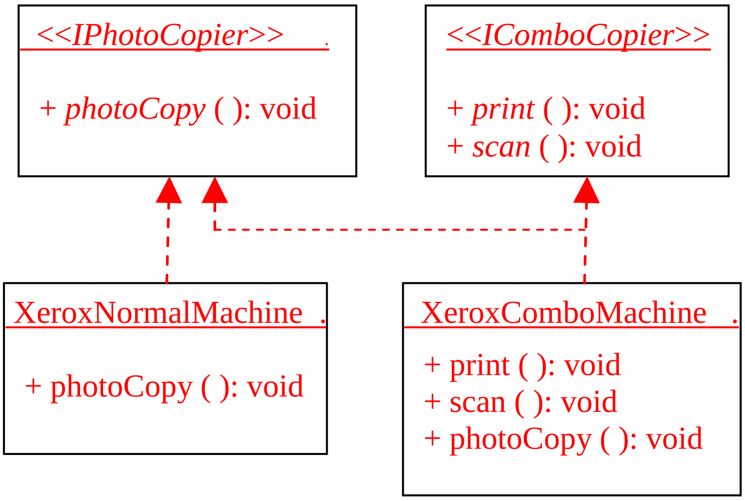
6 marks as shown below

```
public interface IPhotoCopier {
    public void photoCopy();    (1)
}

public interface IComboCopier{(1)
    public void print();         (1)
    public void scan();          (1)
}

public class XeroxNormalMachine
    implements IPhotoCopier {
      public void photoCopy() {
         System.out.println("copy");
      }                                (1)
}
```

```
public class XeroxComboMachine
  implements IPhotoCopier, IComboCopier {
    public void print() {           (1)
      System.out.println("Print");
    }
    public void scan() {
      System.out.println("Scan");
    }
    public void photoCopy() {
      system.out.println("Combo copy");
    }
 }
```

(c) **Draw a UML Class diagram** showing the new design after applying the changes

5 marks for UML


<<IPhotoCopier>>
+ photoCopy ( ): void
<<IComboCopier>>
+ print ( ): void
+ scan ( ): void
XeroxNormalMachine
+ photoCopy ( ): void
XeroxComboMachine
+ print ( ): void
+ scan ( ): void
+ photoCopy ( ): void