CSU Sacramento, Spring 2019 **Dr.Jagan Chidella**

CSC 131-02,03,05 – End Exam SET-B

Student Name:_____

Section: ____

PLEASE WRITE YOUR ANSWERS LEGIBLY

A. TRUE or FALSE - (10*1 = 10 points)

1. REST is an architectural style protocol. True
2. Synchronization is easy in Event-Driven architecture. False
3. Scrum master is ultimately responsible for quality of product outcome. False
4. HTTP Status Code, 4xx is a client error. True
5. A good software has high coupled and less cohesive modules. False
6. Design Patterns describe how to structure classes to meet a given requirement. True
7. In MVC architecture, Model is a request handler. False
8. Use case diagram is a static diagram which models system functionality. False
9. Strongly coupled modules are difficult to change. True
10. Decorator pattern enhances capability of individual object. True

B. Match using ALPHABET only (Choose the best answer) (10*1=10 points)

1. HTTP Status code	C	a) JSON and XML
2. REST HTTP header(s)	J	b) Agile project management tool
3. Standard Formats	A	c) Predefined status of task done at server.
4. Flying Donut	B	d) Independent paths in program.
5. GitHub/GitLab	H	e) Rest Client tool
6. Apache Tomcat	I	f) Reusable solution
7. Cyclomatic Complexity	D	g) Project management tool for handling dependencies
8. Postman	E	h) Version Control service
9. Design Pattern	F	i) Webserver
10. Apache Maven	G	j) Holds meta information

C. Fill in blanks (Total 10 fill in blanks). (10*2 = 20 points)

1. The ____**ITERATOR**_____ pattern provides a way to access the elements of an aggregate object without exposing its underlying representation.
2. Inheritance has higher amount of _____**COUPLING**_____ compared to Composition.
3. Inheritance is __**IS-A**__ relationship and Composition is __**HAS-A**__ relationship. (1 point each)
4. Generators and receivers communicate through an intermediary (called an event queue). This is ____**EVENT-DRIVEN**_____ architecture.
5. REST is an example for _____**CLIENT-SERVER**_____ architecture.
6. Redirecting results/output of an operation into a file instead of output screen is an example for ___**PIPE AND FILTER**_____ architecture.
7. Low cohesion tends to increase ____**COUPLING**_____ between modules.
8. The ___**STRATEGY**_____ design pattern (also known as policy pattern) is a behavioral software design pattern that enables selecting an algorithm at runtime.
9. REST webservices use ___**HTTP**____ protocol as a communication medium between client & server.
10. In ___**FACTORY**_____ design pattern, we create object without exposing the creation logic to client and the client use the same common interface to create new type of object.

Student Name:_____

Section: ____

D. Provide LEGIBLE answers for the following questions. (15*2=30 points)

1. List one advantage and one disadvantage of Layered architecture? (1+1) (1 point in each is expected)
   Advantage:
   a) Improved cohesion which allows for multiple physical deployment options
   b) Decreased coupling which promotes re-use
   c) Specialized expertise
   d) Multiple applicants can reuse the components.
   e) We can test the components independently.
   f) Increases flexibility, scalability, maintainability.
   Disadvantage:
   a) Communicating through multiple layers can be inefficient
   b) Often results in "cheating"

2. Define Code Refactoring and explain its purpose? (1+1)

   The process of changing code without altering its functional behavior.
   Purpose: To improve code for readability/performance/security/compatibility etc.

3. We can instantiate a class without using 'new' operator. Identify which design pattern can do this?
   Factory Pattern

4. List at least 2 components of a REST Response call? (1+1)

   Status Code, Response body, URL, Headers

5. Define Burndown chart and explain its purpose? (1+1)

   It is a graphical representation of work left to do versus time. The outstanding work (or backlog) is often on the vertical axis, with time along the horizontal. That is, it is a run chart of outstanding work.
   Purpose: It is useful for predicting when, the work will be completed.

6. Define stateless protocol? Provide an example. (1+1)

   A stateless protocol does not require the server to retain session information or status about each communicating partner for the duration of multiple requests. Every request is independent of previos request. **Example**: HTTP

7. a) What is the status code returned if GET call is successful? ____**200**_____

   b) Name the header you used to perform Content Negotiation? **Accept or Content-Type**

8. Explain Halstead Complexity?

   Halstead Complexity (calculated from the number of operators and operands).

It reflects the implementation or expression of algorithms in different languages, but be independent of their execution on a specific platform

9. Explain different types of Restful URIs?
   1. Instance Based 2. Collection Based

10. Explain Encapsulation/Information Hiding? What is its purpose?

    Wrapping up of data under a single unit. It is the mechanism that binds together code and the data it manipulates. Variables are made private and methods are made public.

    Purpose: To hide sensitive data/information.

11. Transform the following XML into JSON response

<City>
<id> 1231 </id>
<name> Oakland </name>
</City>
	➔	{
		"id": "1231",
		"name": "Oakland"
		}

12. What is supported as a part of Level 3 HATEOAS?

    Support Responses which contains that a Client can use. Response contains link to access the same resource itself.

13. Identify the type of coupling in following piece of code and provide justification? (1+1)

String city = "oakland";
void main(){
System.out.println(city);
flyDumbo();
System.out.println(city);
}
private void flyDumbo(){
city = "hayward";
}
	➔	Common Coupling.
		Justification: Both the modules use global variables.

14. List at least two advantages of Client Server architecture? (1+1) (Any 2 points are ok)
    a) Centralized control.
    b) Improved cohesion which allows for multiple physical deployment options (e.g., presentation and application logic on one machine; storage on another)
    c) Decreased coupling which promotes re-use

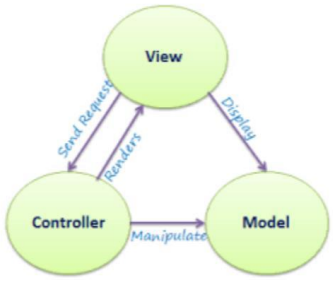
15. Describe
    a) Sprint Backlog Selection Process:
    Backlog grooming ensures that backlog contains the appropriate items, that they are prioritized, and that the items at the top of the backlog are ready for delivery.

    b) Product Backlog grooming:
    Sprint backlog is a task list identified by Scrum team to be completed during Scrum sprint. During sprint planning meeting, the team selects some number of product backlog items (based on priority), usually in form of user stories & identifies tasks necessary to complete each user story.

E. Provide Legible answers for following questions (4*5=20)

1. Explain MVC architecture using the help of a diagram? (Diagram 2.5 + Explanation 2.5)

    - Models – to **realize** problem-domain function
    - View – to **display** data to users
    - Controllers – to **receive and carry out** commands from users



2. Define Coupling and different types of coupling?
    Coupling: Coupling between two modules is a measure of the degree of interdependence or interaction between the two modules.
    Types:
    Data coupling: Modules communicate through a parameter.
    Stamp coupling: Modules communicate using a composite data item.
    Control coupling: Control coupling exists between two modules, if data from one module is used to direct the order of instructions execution in another.
    Common coupling: Two modules are common coupled, if they share data through some global data items.
    Content coupling: Content coupling exists between two modules, if they share code, e.g. a branch from one module into another module.

3. Github work cycle involves following actions (Commit, Modify, Resolve, Pull, Push). Order these actions based on when they would be performed in normal work cycle. You may use an action more than once.

    1. Pull 2. Resolve* 3. Modify 4. Commit 5.Pull 6. Resolve* 7. Push

4. Explain how you may use any Design Pattern for your team project? Provide justification?

   You can possibly use the following Design Patterns

   1. Singleton: Implement this pattern at Server level, to make sure that all the clients/devices get the same instance when they request for an instance.
   2. Observer: Implement this pattern at Server level. Notifications from server are sent to all the clients.
   3. Iterator: Implement this pattern to access lists. Example: Iterate on list of lost devices, found devices etc.
   4. Composite Pattern: A list can hold list of devices. List of lost devices/found devices etc. List exhibits HAS-A relation.

F. Refactor the following code for encapsulation. (5 points)

public class City
{
public String name;
pubic int id;
 	
String getName(){
return name;
}
 	
int getId(){
return id;
}
}
	➔	public class City
		{
		private String name;
		private String id;
		
		public String getName(){
		return name;
		}
		
		public String getId(){
		return id;
		}
		} // Make sure variables are private

G. Refactor the following code to reduce coupling. DO NOT change the functional behavior (4). Explain the type of coupling present in your refactored code (1 points) (4+1=5 points)

String item = “peanut”;
 	
void main()
{
flyDumbo();
}
 	
private static void flyDumbo(){
print(item);
}
	➔	void main()
		{
		String item = “peanut”;
		flyDumbo(item);
		}
		
		private static void flyDumbo(String item){
		print(item);
		}

Coupling Type: **COMMON**		Coupling type present in your code is: **DATA COUPLING**

Student Name:_____

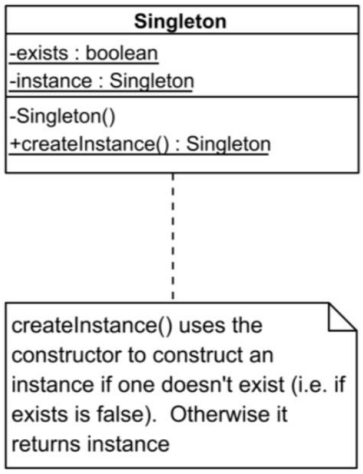
Section: ____

H. Explain Singleton Design Pattern? Draw the UML diagram? List few applications? (3+4+3 = 10 points)

Singleton Pattern: (3 points)

Singleton pattern restricts the instantiation of a class to one "single" instance.
Constructor is made private to restrict instantiation from outside.

UML Diagram: (4 points)



Applications: (2 applications are enough to get 3 points)

1. Windowing systems with one event queue
2. Word processors with one menu bar (for all documents)
3. Hardware Interface access
4. Caching
5. Logger