THEORY QUESTIONS ASSIGNMENT

Software Stream

Name : Shayli Patel

Maximum score: 100

80/100

Maximum score: 100

outstanding work shayli, this was a pleasure to mark!

make sure to go in-depth whereever feasible tho! consider strength & weaknesses of everything, analysing why is it used or what its alternative is

KEY NOTES

- This assignment to be completed at student's own pace and submitted before given deadline.
- There are 10 questions in total and each question is marked on a scale 1 to 10. The maximum possible grade for this assignment is 100 points.
- Students are welcome to use any online or written resources to answer these questions.
- The answers need to be explained clearly and illustrated with relevant examples where necessary. Your examples can include code snippets, diagrams or any other evidence-based representation of your answer.

Theory questions	**10 point each**

1. How does Object Oriented Programming differ from Process Oriented Programming?

Object Oriented Programming (OOP) and Process Oriented Programming (POP) are both programming approaches which use high-level language for programming. OOP is a computer programming model that organizes software design around data, or objects where In POP, a procedural program is typically a list of instructions that execute one after the other starting from the top of the line. In POP, the data security is at risk as data freely moves in the program and code

✓

✓

reusability is not achieved which makes the programming lengthy, and hard to understand. If the task is highly complex, OOP operates better compared to POP. ✓

Below are some key differences between OOP and POP ::

- OOP divides a program into smaller objects, whereas POP divides a program into smaller procedures or functions to arrive at the results of the problem.
- OOP is object oriented and POP structure oriented. ✓
- OOP is known as a bottom-up approach whereas POP is known as a top-down approach.
- In OOP, inheritance property is used in three modes (public private and protected) however there is no provision for inheritance in POP
- In OOP, it uses access specifiers ("public", "private" & "protected") whereas in POP, it doesn't use access specifiers.
- In OOP, encapsulation is used to hide the data. There are three modes - public, private, and protected. hence data security increases. In POP there is no data hiding which means that the data is not secured 8/10
- In OOP, large programs are divided into units called functions where as in POP, the entire program is divided into objects. e.g. go more into depth!

How does OOP work? What is a class, and what is a object explicitly from that?

2. What's polymorphism in OOP?

Polymorphism is one of the core concepts in OOP languages. Polymorphism is the ability to present the same interface for differing underlying forms (data types). It describes the concept that different classes can be used with the same interface. ✓

A real life example of polymorphism: A person at the same time can have

whats the point of polymorphism? advantages / disadvantages etc

different characteristics. For example, a woman can be a sister, daughter, friend and an employee. So the same women would possess different behavior in different situations.

5/10

answer is unfortunately brief - can you go more in-depth?

With polymorphism, each of these classes will have different underlying data.

how can polymorphism be implemented? e.g. overriding or overloading

outside of real-life instances, can you give a programming example e.g. two classes that inherit & override

3. What's inheritance in OOP?

Inheritance is another core concept of object-oriented programming (OOP) languages. It is a mechanism where you can derive a class from another class for a hierarchy of classes that share a set of attributes and methods.

Inheritance is meant to optimize the work of programmers. The role that inheritance plays in this optimization is in allowing software engineers to create class hierarchies, where classes and objects inherit properties and behaviors from their parent class. A class that inherits from a parent class is called a subclass or child class, and objects that receive properties and behaviors from a parent through inheritance are referred to as child objects.

can you give examples / more depth?

7/10

usability of inheritance (e.g. code re-use)?

adv / disadv of inheritance?

4. If you had to make a program that could vote for the top three funniest people in the office, how would you do that? How would you make it possible to vote on those people?

Create list of numbered employees and print the list so the voters only input the corresponding number .

```
Staff = [‘1. Shayli’, ‘2. Chris’, ‘3. Jess’]

for member in staff:

    print(member)
```

Create an empty dictionary

```
votes = {}
```

Use a function that adds a key (name) to the dictionary or add a vote to an existing key (name).

Then use an input function for voters to input the employee number.

```
def add_votes(votes):
    vote = input('Who would like to vote for? ')
    if "a" in a_dict:
        votes[vote] += 1
    else:
        votes[vote] = 1
```

what is "a" here? Shouldn't it be technically 'vote'?

since vote is the input that we're voting for / increasing count of potentially

isnt example code wrong potentially?

✓ 9/10

Then use the sorted function on the dictionary to sort the keys by highest values.

You can then print off the three most voted staff and how many votes they got.

5. What's the software development cycle? ✓

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality softwares. The SDLC aims to produce high-quality software that meets or exceeds customer expectations, reaches completion within set timelines and cost estimates. SDLC is a framework defining tasks performed at each step in the software development process. The SDLC is continuous as once software has been deployed, the team will move on to planning in the next cycle.

Stage 1 - Planning and Requirement Analysis

Stage 2: Defining Requirements ✓

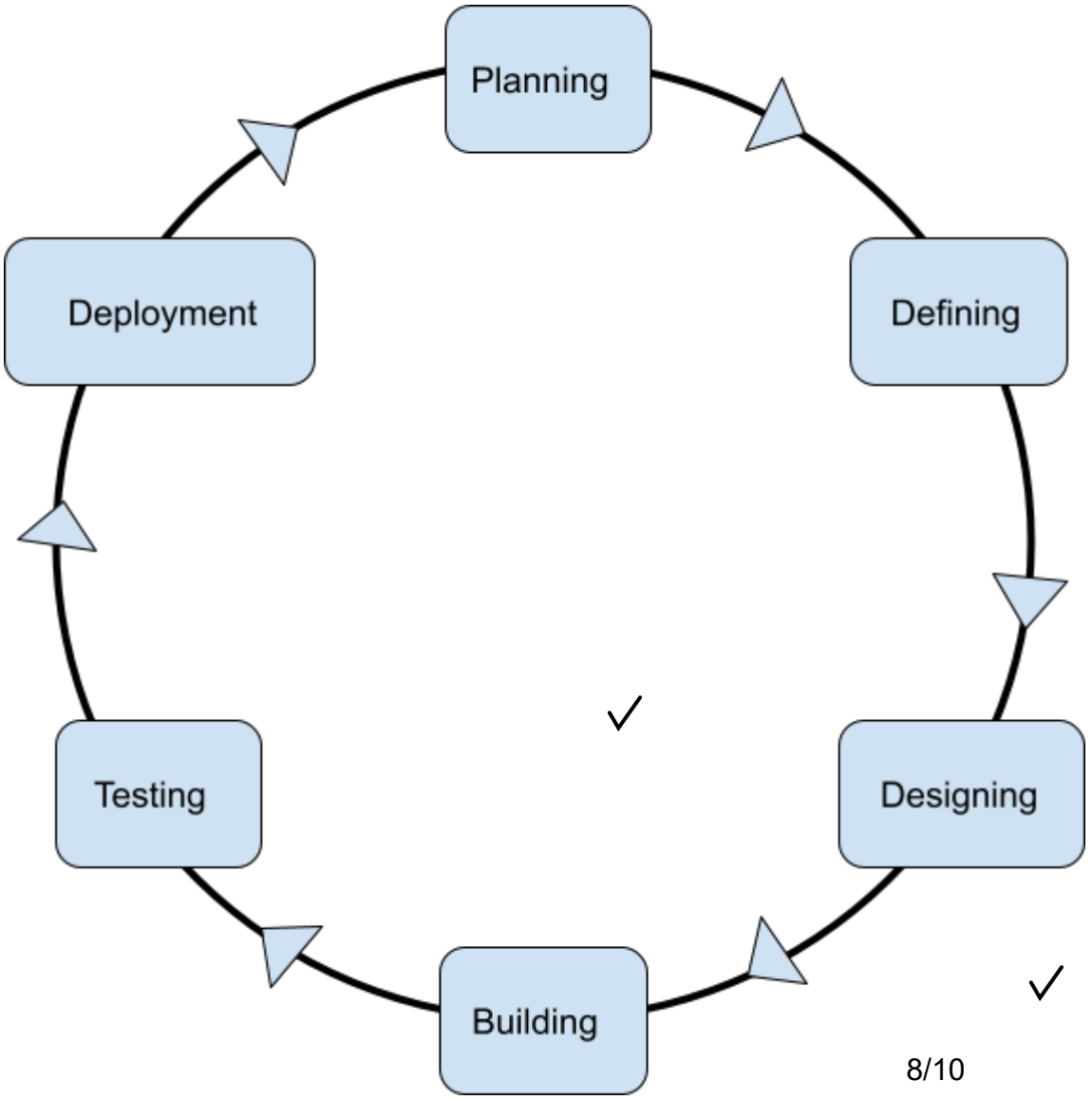
Stage 3: Designing the Product Architecture

Stage 4: Building or Developing the Product ✓

Stage 5: Testing the Product

Stage 6: Deployment in the Market and Maintenance

what is each stage meant for (even if its implicit) - for example, what is a requirement? What is meant by the product architecture (its tech stack)?



8/10

what are the strength / weaknesses associated with such a structured process? basically be more critical if possible - who uses this structured process, why?

There are various software development life cycle models defined and designed which are followed during the software development process. For example: Waterfall model , Agile model, Iterative model, Spiral model , or V-Model

6. What's the difference between agile and waterfall?

Agile development is a team-based approach that emphasizes rapid deployment of a functional application with a focus on customer satisfaction. It defines a time-boxed phase called a sprint with a defined duration of two weeks.

Waterfall project management is a sequential approach that divides the SDLC to distinct phases such as requirements gathering, analysis and design, coding and unit testing, system and user acceptance testing, and deployment. The next phase can only proceed if the previous phase has been completed. In between phases, a deliverable is expected or a document is

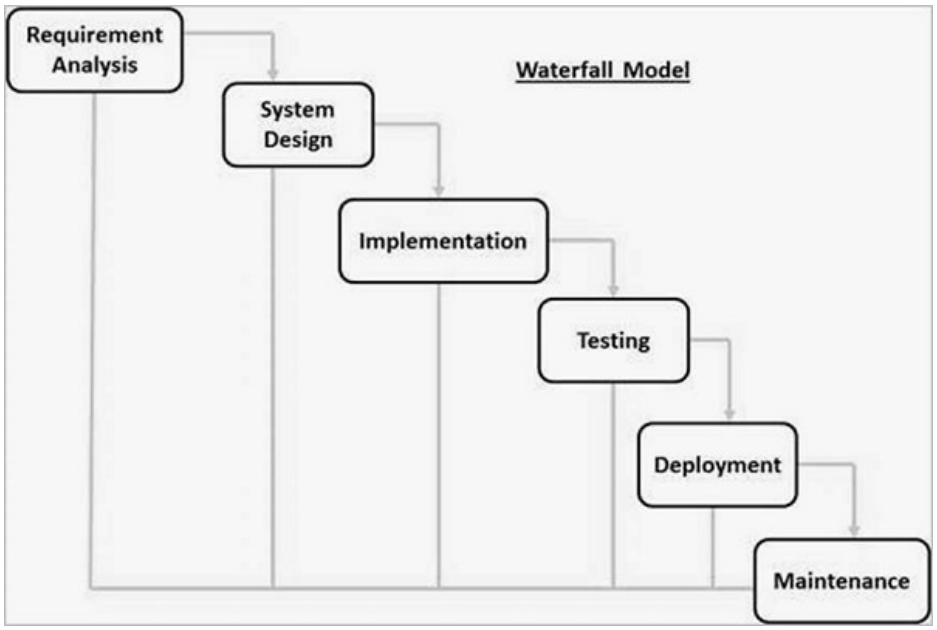
signed off.

- Agile is an incremental and iterative approach whereas Waterfall is a linear and sequential approach. ✓
- Agile separates a project into sprints and Waterfall divides a project into phases.
- Agile helps complete many small projects whereas Waterfall helps complete one single project. ✓
- Agile introduces a product mindset with a focus on customer satisfaction and Waterfall focuses on successful project delivery. ✓
- Requirements are prepared everyday in Agile, while requirements are prepared once at the start in Waterfall.
- Agile allows requirement changes at any time whereas Waterfall avoids scope changes once the project starts. ✓
- Testing is performed concurrently with development in Agile however in a Waterfall project, testing phase comes only after the build phase.
- Test teams in Agile can take part in requirements change whereas test teams in Waterfall do not get involved in requirements change
- Agile enables the project team to operate without a dedicated project manager but Waterfall requires a project manager who plays an essential role in every phase. ✓

impact wise, whats the difference? is one better than the other?

See diagram of the waterfall model below:

why would someone use waterfall over agile?



if feasible go into more depth - who uses waterfall, who may alternatively use agile? what makes one more suitable than the other
e.g. start-ups who are product-focused may use agile more

✓

alternatively, waterfall is worse since you can't go back easily (e.g. in case you discover a massive requirement mistake) - would agile be better than for xyz?

8/10

7. What is a reduced function used for?

The reduce() function in Python is a function that implements a mathematical technique called

✓

folding or reduction. reduce() is used when you need to apply a function to an iterable and reduce it to a single cumulative value.

It takes an existing function, applies it cumulatively to all the items in an iterable, and generates a single final value. It can be used for processing iterables without writing explicit for loops. ✓

reduce() was originally a built-in function in Python 2.x but was moved to functools library in Python 3.0 due to performance and readability issues. When using Python 3.0+, the functools library needs to be imported and then use fully-qualified names like functools.reduce().

Below is an example of a function that computes the sum of two numbers and prints the equivalent math operation. ✓

```
from functools import reduce

def add(a,b):

    result=a+b

    print('{} + {} = {}'.format(a, b, result))

    return result

numbers = [0, 1, 2, 3, 4]

reduce(my_add, numbers)
```

✓

what is the 3rd parameter of the reduce function
e.g. what is it called, its purpose, etc

Output

```
0 + 1 = 1

1 + 2 = 3

3 + 3 = 6

6 + 4 = 10
10
```

10/10

8. How does merge sort work ✓

Merge sort is one of the most efficient sorting algorithms which works on the principle of Divide and Conquer. Merge sort repeatedly breaks down a list into several sublists until each sublist consists of a single element and merges those sublists in a manner that results in a sorted list.

There are two types of merge sort approaches:

- The **top-down merge sort approach** is the methodology which uses recursion mechanism. It starts at the Top and proceeds downwards, with each recursive turn asking the same question such as “What is required to be done to sort the array?” and having the answer as,

✓ 8/10

consider:

why would someone use merge sort over other sorting algorithms?
in essence, what is the performance cost of merge sort (time, space)? Why is this good / bad

how does divide & conquer help / why is this approach good for performance?

"split the array into two, make a recursive call, and merge the results.", until one gets to the bottom of the array-tree.

- The **bottom-Up merge sort approach** uses iterative methodology. It starts with the "single-element" array, and combines two adjacent elements and also sorting the two at the same time. The combined-sorted arrays are again combined and sorted with each other until one single unit of sorted array is achieved.

9. Generators - Generator functions allow you to declare a function that behaves like an iterator, i.e. it can be used in a for loop. What is the use case?

Generators allow programmers to make an iterator in a fast, easy, and clean way.

Generators introduce the yield statement to Python and works similar to the return statement because it returns a value.

By using a generator, we can restrict CPU cycles to only the elements we really need and will only yield the elements we specified. Instead of the amount of "useless CPU cycles" spent on generating an entire list will be significantly reduced.

Another potential benefit of using generators is saving memory, given we do not need all elements of the generator in memory concurrently so generators will be used when memory is limited. For example if an element isn't needed, the space that object "occupied" can be recycled and used for the next element.

10/10

can you give an example? like wha tthe code may look like

10. Decorators - A page for useful (or potentially abusive?) decorator ideas. What is the return type of the decorator?

A decorator is a callable function or class that accepts either a function or a class and returns a new function or class that wraps around the original one. Decorators use a special decorator syntax (@) before the decorator name and is placed above the function that it is being wrapped around.

A decorator is a function that accepts a function and returns a function.

It can also be a class which accepts a function and returns an instance of that class (a "decorator class" that can be used as a "function decorator").

A decorator can also be a function which accepts a class (a "decorator function" that can be used as a "class decorator").

why would a programmer use decorators though?

what's the point of it - why not modify existing functions to be more advanced and hence meet whatever utility that a decorator can bring?

can you give code examples as well or make reference to one e.g. flask

7/10