

THEORY QUESTIONS ASSIGNMENT

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

Object Oriented Programming (OOP) is a programming language that focuses on the object that is to be manipulated. The objects are collected and arranged in a way that translates their relations to each other. This is called data modelling. On the other hand, Process Oriented Programming (POP) is a programming language that follows an approach that breaks down tasks into variables and routines. The main program is divided into smaller parts called functions and these can be treated as individual smaller programs. POP is not ideal for complex problems because its structure makes it difficult to organise the code. However, OOP is designed in a way that allows different parts of the program to relate to each other due to its structure. OOP has classes, objects, and methods as a structure. Classes are data type that act as blueprints for the objects. Objects are instances of the class created and they can be real-life objects or abstract entities. Methods are the functions that are defined inside the class.

Some of the main principles of OOP includes; encapsulation, abstraction, inheritance, and polymorphism. These principles bring benefit to the programming. For example, encapsulation allows bundling of data and the mechanism of restricting the access of some components of the object. Other objects don't have access to this information and can't make modifications to it. This is beneficial when using for security and privacy purposes. POP does not have encapsulation and therefore the data is less secure. POP does not have inheritance and polymorphism. It focuses more on the process rather than the object. This means that functions in POP cannot relate to each other in the same way it does in OOP.

Some programming languages that use OOP are: Java, Python, Javascript, C#

Some programming languages that use POP are: C, Pascal, VB

2. What's polymorphism in OOP?

Polymorphism is a core principle of OOP. It refers to the use of single type entity to represent different types. For example the inbuilt function, `len()`, remains the same but can be used for strings, lists, and dictionaries.

```
print(len("Hello"))
print(len(["Python", "JS", "React"]))
print(len({"Name": "Sam", "Address": "India"}))
```

Output:

```
5
3
2
```

Polymorphism is also implemented in classes. Python allows different classes to have same method names. These classes are polymorphic because they allow a single interface to

interact with entities of different types. In return, python will run through the program and output the each method with their respective class.

```
class Rabbit():
    def age(self):
        print("This function determines the age of Rabbit.")

    def colour(self):
        print("This function determines the colour of Rabbit.")

class Horse():
    def age(self):
        print("This function determines the age of Horse.")

    def colour(self):
        print("This function determines the colour of Horse.")

obj1 = Rabbit()
obj2 = Horse()
for type in (obj1, obj2):
    type.age()
    type.colour()

This function determines the age of Rabbit.
This function determines the colour of Rabbit.
This function determines the age of Horse.
This function determines the colour of Horse.
```

This is an example of how polymorphism can be used with classes. As evident, a loop is created that iterates through the objects and the methods are called. This provides greater flexibility when using OOP.

3. What's inheritance in OOP?

Inheritance allows us to define a class (child) that receives all its functionality from a parent class and add additional information. The parent class hold the base information and the child class has more specific information. Inheritance is a core principle in OOP. It accurately represents real-world relationships and allows the reusability of a code. Method overriding in python makes it possible for the child class to override the parent class. The `__init__()` is defined in both classes so that the method in the derived class can overtake the parent class.

An example of inheritance in OOP is:

```
class Person(object):

    def __init__(self, name):
        self.name = name

    def getName(self):
        return self.name
```

```
def isEmployee(self):  
    return False  
  
class Employee(Person):  
  
    def isEmployee(self):  
        return True  
  
# Driver code  
emp = Person("Geek1")  
print(emp.getName(), emp.isEmployee())  
  
emp = Employee("Geek2")  
print(emp.getName(), emp.isEmployee())  
  
Output:  
Geek1 False  
Geek2 True
```

This example shows that the class Employee is a child class of the Person class. It inherits all the qualities of the parent class and adds onto that with an additional method that checks if the person is an employee. Inheritance is useful in OOP because it can be re-use, the program structure is readable, and it makes it more flexible to changes.

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 a variable for each of the nominated people and assign an input string 'enter the nominee name'. We then create an integer variable to count the number of votes but assign the value 0 to the all for a clean start. Create a while loop and inside the loop create if statements for each nominated funny person. In the if statements, add a vote for each time the respective nominee gets a vote. Close the loop and place the values of the votes in a dictionary. Then using the range(len(list) functions, create a function that will sort the values in the dictionary in ascending order.

5. What's the software development cycle?

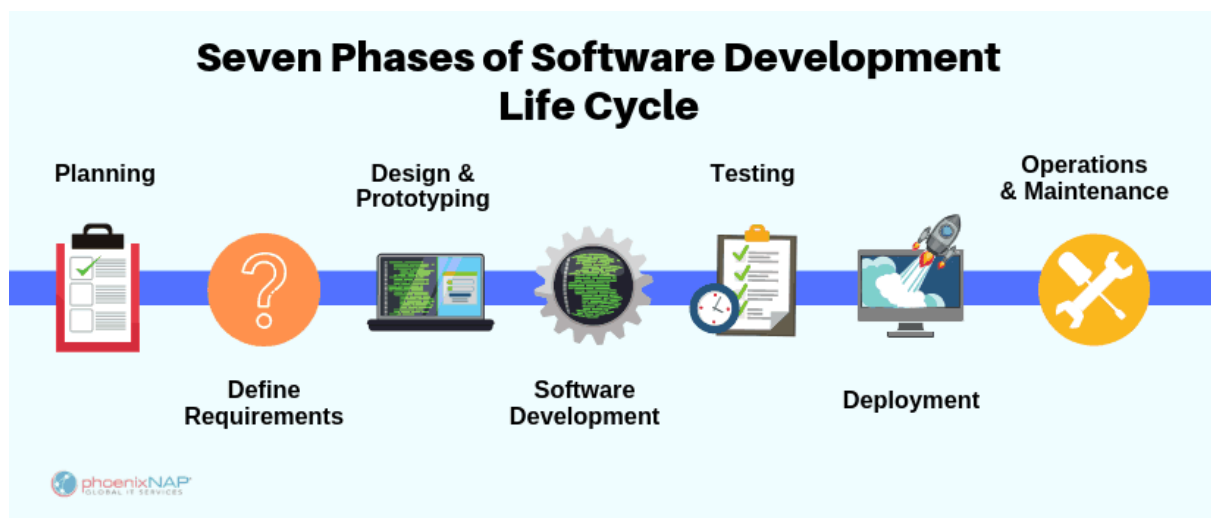


Figure 1: The seven stages of the software development cycle.

Software development cycle (SDC) is used to measure and improve the development process. Efficiency at each stage is maximised by outlining the required tasks and focusing on that at each stage.

The planning stage involves the project leaders defining the project outline and scope. The logistics of the projects and time frame are discussed and finalised.

During the requirement stage, the requirements of the project and process are established.

The design phase of the project focuses on the way the application will work. The architecture and user interface are discussed and designed. Generally, a prototype of the project is created.

Software development is the 4th stage of the software development cycle. This is when the actual programming of the application takes place. The application is split into sections and developers are assigned sections.

Once the software development is done (and sometimes during it) the testing takes place. Testing is a crucial stage in the SDC. Before the application is released, it gets tested for bugs, that functions work properly, and reduce the number of glitches. QA team members are in charge of testing the functionality of the program.

The second to last stage is deployment. In the deployment stage, the application is made available to the users. The final stage is maintenance. After the application is deployed, it goes under maintenance. In this phase, users discover bugs that weren't found in testing and is solved.

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

Waterfall methodology is the classic way of development. When each stage of the development finishes, the project moves onto the next. For example, testing will happen only after the development is finished. This method slows down production because each phase relies on the completion of the other. When waterfall methodology was introduced, software systems were more complicated and required documentations. This meant that the waterfall methodology worked well because products were not needed to be released as often as they are now. However, in the early 2000's, technology became less complicated and the waterfall methodology was becoming outdated. Requiring documentations, specifications, and doing a big release at the end slowed down progress. This is when the agile methodologies were introduced.

Agile methodology, on the other hand, works by building products using short cycles of work. It focuses on releasing software cycles quickly and then asking for user feedback. The core focus of the process is the users. User stories are created, and the product is built on those. The product owner is the one that is in charge of the product and the satisfying the user's requirements. He or she will break down the product vision and be in constant communication with the developers to ensure they are meeting the needs of the users. The software development team are the ones who write the program. They focus on a few tasks and deliver those at a time.

The most popular agile methodology is called scrum. It works by having sprints. These sprints last around two weeks and focuses on releasing a product in the short frame of time. The structure of a scrum is that of the following:

- Sprint Planning Meeting
- Daily Scrum Meeting
- Sprint Review Meeting
- Sprint Retrospective Meeting

In conclusion, agile methodologies work better in software development today as products are constantly evolving and require new feature updates. This methodology allows user inputs and prevents product *releases taking long*.

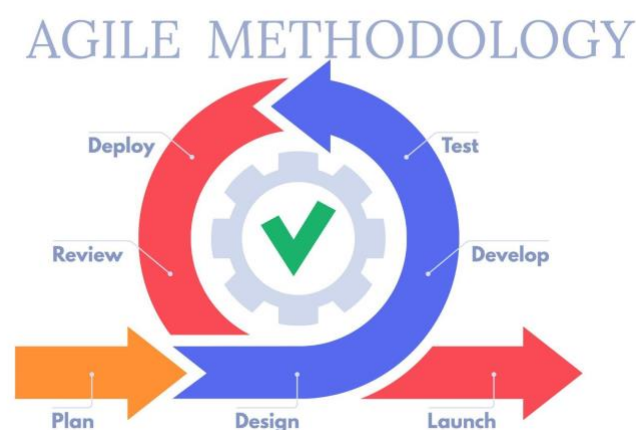


Figure 2: Agile methodology cycle

7. What is a reduced function used for?

The reduce function is used when you need to apply a function to an iterable and reduce it to a single cumulative value. The purpose of this is to reduce the iterable to a single cumulative value without writing a for loop. The reduce function has two arguments; function and iterable. The function argument will be applied to the iterables to output a final value. The second argument is an iterable. This could be lists, sets, dictionaries, or tuples.

```
from functools import reduce

def add(a, b):
    result = a + b
    print(f"{a} + {b} = {result}")
    return result

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

reduce(add, numbers)
```

Output:
 0 + 1 = 1
 1 + 2 = 3
 3 + 3 = 6
 6 + 4 = 10

Reduce function will apply the add function to the first to items in the number variable and output a result. Then it will move to the next items and add on to the initial result. It will continue until it loops over the whole list. The final result is 10.

8. How does merge sort work ?

Merge sort is an algorithm that sorts values in a data structure. It works by splitting the values into halves, sorting it, and then merging them back together.

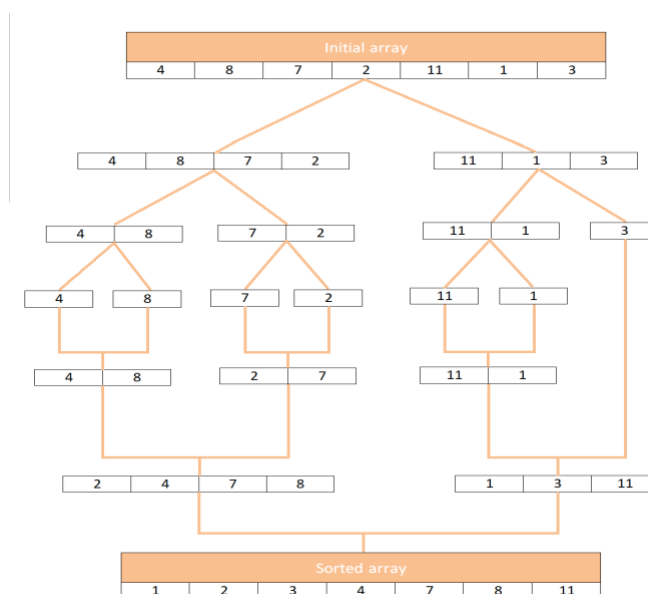


Figure 3: Diagram of top-down merge sort approach.

As illustrated in the diagram, the list is divided into sub-lists, sorted and then merged together and then finally merged to into a sorted list. There are two approaches to using merge sort: top-down and bottom-up. Bottom-up approach sorts the subarrays of one element then merges the result into sub-lists of two elements and continues until the full list is sorted. Top-down approach is shown in figure 3, where the list is divided into sub-lists and then sorted before merging back into the final sorted list.

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?

Generator functions are simple ways of creating iterators. Its iterator objects with a sequence of values using a yield statement. When the generator function reaches the 'yield' statement, it outputs that value.

```
def generator_function():  
    for i in range(10):  
        yield i  
  
for item in generator_function():  
    print(item)
```

Generators are best used when dealing with large sets of results where the calculations may have to loop over each other.

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

Decorators are used to add functionality to a code. Similar to the reduce function, decorators are functions that can be passed as arguments to another function. Decorators don't alter the function code but rather add a layer of functionality to it. The @ symbol is used and is placed above the function.