1- What are OOPs?

To be honest I don’t know what the plural “s” is for
but I will answer with my understanding of OOP.

OOP stands for “Object Oriented Programming”.
it’s the style of programming that is based on the concept of partitioning the program to entities called “Objects”, those objects come from something we call class, which is the blueprint that defines functionality and attributes of every object made with that class, this Object Oriented style of programming also includes another concepts such as abstraction, polymorphism, composition and inheritance.

2- What are the differences between an interface and an abstract class?

Simple answer is: the abstract class is a class and interface is not a class it’s an interface.

long answer: an abstract class is a class with some sort of abstraction in the functionality, depending on the programmer, so the programmer can write some functionality in some methods or functions, and leave the rest functions abstracted without any functionality on the base class.

while on the other hand an interface is not a class, you can’t write any functionality in it, pure abstraction without any functionality on it’s own.

another difference is that in java, I don’t remember if it’s the same in any programming languages, a class can implement multiple interfaces, while it can only extend one class “in java we use implement keyword to inherit an interface and extend to inherit an abstract class”.

3- What are differences between private, protected, and public access modifiers?

Private variables and functions: only accessed inside the same class.

Protected variables and functions: accessed in the same class, same package and in a subclass.

Public variables and functions: can be seen accessed everywhere when called.

4- What is the difference between overriding and overloading?

Overloading is two functions within the same class and with the same name, but with different parameters.

overriding is a different implementation provided in the subclass for a function in the baseclass and must use the same parameters.

5- What is the difference between an error and an exception?

An error can’t be caught and checked, if it happens it drives the app to crash, that’s the only option, you can define it change the message that is shown when it happens, but it would still stop the app from working.

While an exception should be caught in a try-catch block and the app will be still running without problems.

6- What are SOLID principles? And give an example for 3 principles

SOLID principles: a set of rules in Object Oriented design, that makes the code easier to maintain, modify and understand.
S: Single Responsibility Principle: one reason to change the class, so if a class has two responsibilities and two reasons to be changed, then functionality should be splitted into two classes.

O: Open-Closed principle: the code must be open to extinction but closed for modification, which means the code that works should be untouched, and any new code should be extensions of the original work.

L: Liskov Substitution principle: subclasses should never break the parent class type definition.

7- What's BigO?

An estimated description for the complexity and running time for a set of code based on the number of times commands are commited.