

# OOPs

## Assignment Solutions



## Q1. What are the differences between class and object ?

A1. Differences between class and object are following

Class	Object
Class is a blueprint from which objects are created.	Object is an instance of a class.
Class is a logical entity.	Object is a real world entity such as pencil, laptop etc.
Class is created using the class keyword.	Object is a physical entity.
Class is declared only once.	Object can be created many times.

## Q2. Create a class named as Student having attributes like String name, String rollNumber and String mobileNumber with a constructor accepting all the three attributes.

A2. Student class is created with name, rollNumber and mobileNumber as attributes.

Code link: <https://pastebin.com/BQjFyzq7>

## Q3. Create an object of the Student class created above and print its attribute values as assigned by you.

Code link : <https://pastebin.com/CskLgTNS>

Output:

```
Aman 4675 8978988767
...Program finished with exit code 0
Press ENTER to exit console.█
```

## Q4. Why do we need a constructor ? Explain with the help of an example.

A4. We need a constructor because constructor is required to create an object of the class i.e an instance of the class. In the below code, if we had not defined a constructor or if C++ would not have initialized a default constructor then the code will not be able to create an instance of the class, resulting in a compile-time error.

Code link: <https://pastebin.com/mQa3kSfR>

## Q5. Make a No-argument constructor of the class named Book which has attributes of string name and integer id and check output by printing the attribute values of the object after initializing it.

Code Link : <https://pastebin.com/mQa3kSfR>

```
Constructor called
0

...Program finished with exit code 0
Press ENTER to exit console.█
```

**Q6. What are the benefits of using getter and setter methods to access private instance variables?**

**Ans.** Getter and setters are functions that are used to access private data members of the class and perform required operations on them. The benefits of using getter and setter methods to access private instance variables include improved security (since the variables cannot be directly accessed from the outside), better control over the values that can be assigned to the variables (through the use of validation checks in the setter methods), and the ability to update other parts of the program or perform other actions when the variables are changed (through the use of code in the setter methods).

**Q7. How does inheritance allow you to reuse code in a program?**

**Ans.** Inheritance allows you to reuse code in a program by allowing you to create a new class that is based on an existing class. The new class, called the subclass, automatically inherits all the attributes and behaviors of the existing class, called the superclass. This allows you to create a hierarchy of classes, where each subclass adds or modifies the attributes and behaviors inherited from the superclass.

**Q8. How do you use encapsulation and inheritance to design and implement a program that is secure, robust, and scalable?**

**A4.** To design and implement a program that is secure, robust, and scalable using encapsulation and inheritance, you can follow these best practices:

1. Use encapsulation and abstraction to hide the implementation details of your classes from the outside world. This can help to improve security by preventing unauthorized access to sensitive data or operations, and it can also make your program more robust by minimizing the chances of unintended side effects when the implementation changes.
2. Use getter and setter methods to access private instance variables, rather than accessing them directly. This can help to improve security by providing a way to validate or transform the values that are assigned to the variables, and it can also make your program more robust by allowing you to update other parts of the program or perform other actions when the variables are changed.
3. Use inheritance to reuse code and create a hierarchy of classes. This can help to improve scalability by allowing you to write less code and make it easier to add new features to your program. It can also make your program more flexible and easier to maintain by allowing you to modify the behavior of a subclass without affecting the superclass.
4. Use abstract classes and methods to define the interface of a class without providing a complete implementation. This can help to improve scalability by allowing you to create a reusable base class that can be extended by multiple subclasses, and it can also make your program more flexible and easier to maintain by allowing you to change the implementation of the abstract class in one subclass without affecting the other subclasses.

**Q9. Give a function overloading C++ code on the topic 'life'.**

**Ans.** Here is the C++ code implementing function overloading on the topic 'life'. We have calculated age in years, months and days by passing different parameters here.

Code link: <https://pastebin.com/QGPdRfNA>

**Q10. Write a Class with a function to add squares of two private numbers and return it to the user without using getter and setter.**

**Ans.** Here we need to add squares of two private numbers so we will use a friend function to implement the following operation.

Code link: <https://pastebin.com/i3eAfctC>

169

```
...Program finished with exit code 0
Press ENTER to exit console.[]
```