1• WAP to print “Hello World” using C++?

```
#include <iostream>

using namespace std;

int main() {

  cout << "Hello, World!" << endl; // This prints Hello, World!

  return 0;

}
```

2. • What is OOP? List OOP concepts?

As the name suggests, Object-Oriented Programming or OOPs refers to languages that use objects in programming, they use objects as a primary source to implement what is to happen in the code. Objects are seen by the viewer or user, performing tasks assigned by you. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism etc. in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.
Let us discuss prerequisites by polishing concepts of method declaration and message passing. Starting off with the method declaration, it consists of six components:

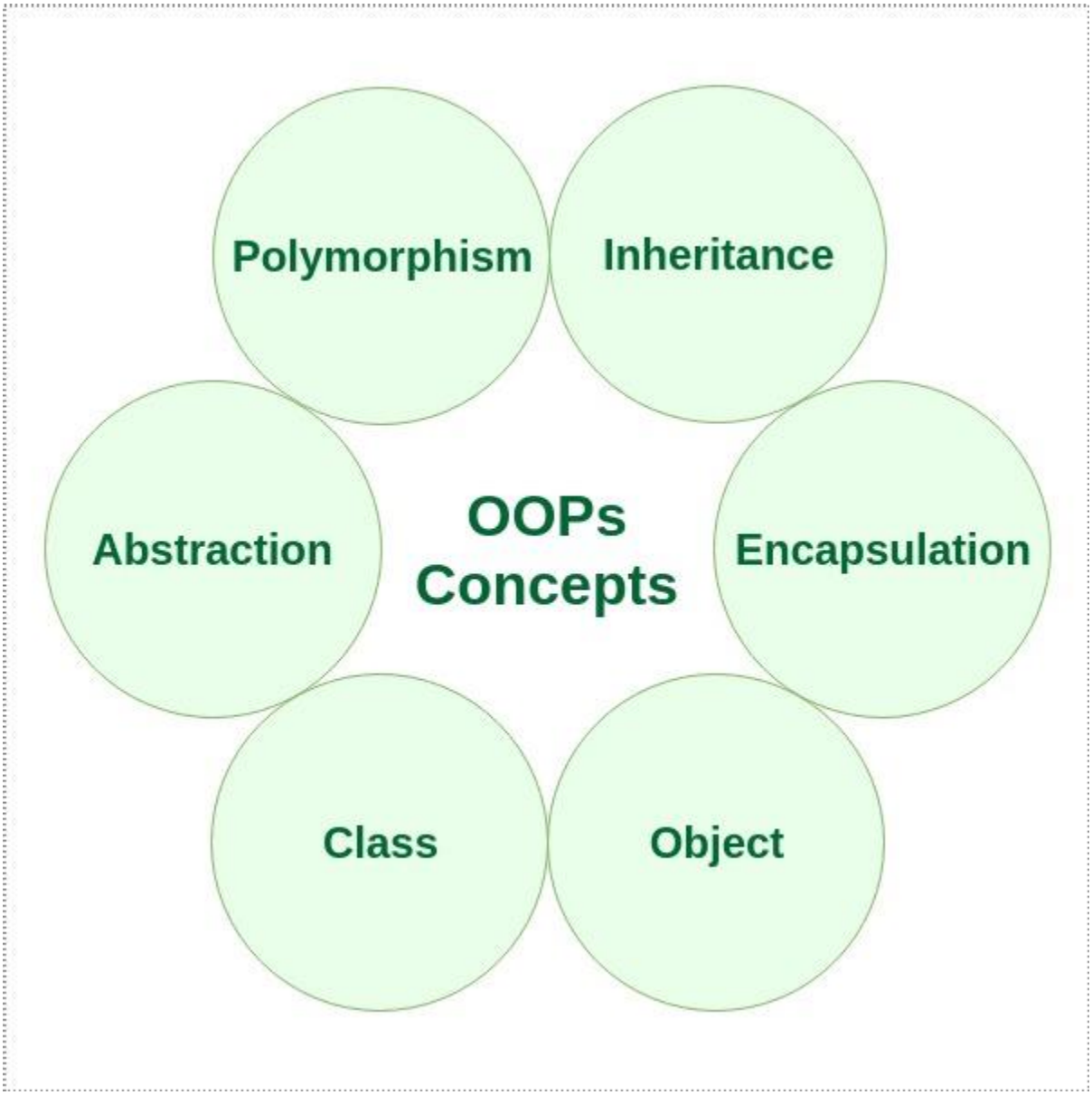
- **Access Modifier**: Defines the **access type** of the method i.e. from where it can be accessed in your application. In Java, there are 4 types of access specifiers:
    - **public:** Accessible in all classes in your application.
    - **protected:** Accessible within the package in which it is defined and in its **subclass(es) (including subclasses declared outside the package)**.
    - **private:** Accessible only within the class in which it is defined.
    - **default (declared/defined without using any modifier):** Accessible within the same class and package within which its class is defined.
- **The return type**: The data type of the value returned by the method or void if it does not return a value.
- **Method Name**: The rules for field names apply to method names as well, but the convention is a little different.
- **Parameter list**: Comma-separated list of the input parameters that are defined, preceded by their data type, within the enclosed parentheses. If there are no parameters, you must use empty parentheses ().
- **Exception list**: The exceptions you expect the method to throw. You can specify these exception(s).

- **Method body**: It is the block of code, enclosed between braces, that you need to execute to perform your intended operations.

Message Passing: Objects communicate with one another by sending and receiving information to each other. A message for an object is a request for execution of a procedure and therefore will invoke a function in the receiving object that generates the desired results. Message passing involves specifying the name of the object, the name of the function and the information to be sent.
Now that we have covered the basic prerequisites, we will move on to the 4 pillars of OOPs which are as follows. But, let us start by learning about the different characteristics of an Object-Oriented Programming Language.

OOPS concepts are as follows:

1. Class
2. Object
3. Method and method passing
4. Pillars of OOPs
    - Abstraction
    - Encapsulation
    - Inheritance
    - Polymorphism
        - Compile-time polymorphism
        - Runtime polymorphism



A class is a user-defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. Using classes, you can create multiple objects with the same behavior instead of writing their code multiple times. This includes classes for objects occurring more than once in your code. In general, class declarations can include these components in order:

1. **Modifiers**: A class can be public or have default access (Refer to this for details).
2. **Class name:** The class name should begin with the initial letter capitalized by convention.
3. **Superclass (if any):** The name of the class's parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.

4. **Interfaces (if any):** A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.
5. **Body:** The class body is surrounded by braces, { }.

An object is a basic unit of Object-Oriented Programming that represents real-life entities. A typical Java program creates many objects, which as you know, interact by invoking methods. The objects are what perform your code, they are the part of your code visible to the viewer/user. An object mainly consists of:

1. **State**: It is represented by the attributes of an object. It also reflects the properties of an object.
2. **Behavior**: It is represented by the methods of an object. It also reflects the response of an object to other objects.
3. **Identity**: It is a unique name given to an object that enables it to interact with other objects.
4. **Method:** A method is a collection of statements that perform some specific task and return the result to the caller. A method can perform some specific task without returning anything. Methods allow us to **reuse** the code without retyping it, which is why they are considered **time savers**. In Java, every method must be part of some class, which is different from languages like C, C++, and Python.

3. What is the difference between OOP and POP?

Object-Oriented Programming (OOP)

- OOP treats data as a critical element in the program development and does not allow it to flow freely around the system.
- In OOP, the major emphasis is on data rather than procedure (function).
- It ties data more closely to the function that operate on it, and protects it from accidental modification from outside function.
- OOP allows decomposition of a problem into a number of entities called objects and then builds data and function around these objects.
- The data of an object can be accessed only by the function associated with that object. However, function of one object can access the function of other objects.
- C++, Java, Dot Net, Python etc are the example of Object oriented programming (OOP) language.

Some Characteristics of Object Oriented Programming:-

- Emphasis is on data rather than procedure (function).
- Programs are divided into objects.

- Functions that operate on the data of an object are ties together in the data structure.
- Data is hidden and cannot be accessed by external function.
- Objects may communicate with each other through function.
- New data and functions can be easily added whenever necessary.
- Follows bottom up approach in program design.

Features of OOP (Object Oriented Programming):-

1. Class
2. Object
3. Encapsulation
4. Data Abstraction
5. Inheritance
6. Polymorphism
7. Data binding
8. Message Passing

Application of Object Oriented Programming:-

- User interface design such as windows, menu
- Real Time Systems such as Control system for cars, aircraft, space vehicles etc
- Office automation system such as Document Management System i.e. Word processing system, spread sheet software etc
- AI and Expert System
- Neural Networks and parallel programming System
- Decision support system etc

Advantages of OOP:-

1. Improved software-development productivity:
Object-oriented programming is modular, as it provides separation of duties in object-based program development. It is also extensible, as objects can be extended to include new attributes and behaviors. Objects can also be reused within an across applications. Because of these three factors – modularity, extensibility, and reusability – object-oriented programming provides improved software-development productivity over traditional procedure-based programming techniques.

2. Improved software maintainability:
Since the design is modular, part of the system can be updated in case of issues without a need to make large-scale changes.

3. Faster development:
Reuse enables faster development. Object-oriented programming languages come with rich libraries of objects, and code developed during projects is also reusable in future projects.

4. Lower cost of development:
The reuse of software also lowers the cost of development. Typically, more effort is put into the object-oriented analysis and design, which lowers the overall cost of development.