

Chapter 2 Answers

1. Static and dynamic Polymorphism.

- **Static Polymorphism:** it refers to the ability of the compiler to relate or bind the function call with function definition during compilation itself. There are two methods for static Polymorphism, function overloading and operator overloading.
- **Dynamic Polymorphism:** It refers to binding the function definition with function call during runtime.

2. (a) Function Overloading

```
(b) #include<iostream>
using namespace std;
float area(float radius) {
    return 3.14 * radius * radius;
}
int area(int length, int breadth) {
    return length * breadth;
}
int main() {
    cout << "Area of Circle: " << area(5.0f) << endl;
    cout << "Area of Rectangle: " << area(4, 6) << endl;
    return 0;
}
```

3. Data Abstraction vs Data Encapsulation:

- **Data Abstraction:** Hiding details from outside world and shows only essential features.
- **Data Encapsulation:** Binds data and functions into a single unit (class).

4.

Class	Structure
Contain data and function as members.	Contain only data as members.
Access specifiers private, public and protected are used for members.	No Access specifiers are used
By default, the members are private	By default, the members are public
Class declared using keyword class	Structure declared using keyword struct

5. (a) private

6. Data Abstraction

7. Object oriented programming binds data and function into a single unit called object.

Advantages:

- Easy to maintain and modify code
- Provides modular structure for programs

8. Polymorphism

9. Procedural oriented programming consists of a set of instructions and organizes these instructions into a function.

Disadvantages:

- Data is undervalued
- Adding new data elements need modification to functions
- Difficult to create new data type
- Provide poor real world modelling

10. Data Encapsulation

11. It is the ability to process objects differently depending on their data type or class.

Eg: `int add(int a, int b);`

`float add(float a, float b);`

12.

Procedural oriented paradigm	Object oriented Paradigm
Data is undervalued	Data is given importance
Procedure is given importance	Procedure is driven by data
Creating new data type is difficult	New data type and associated operations can easily be defined
Poor real world modelling	Easy to define real world scenarios
programs are divided into functions	programs are decomposed into object

13. It is the ability to process objects differently depending on their data type or class.

Types:

- Compile-time polymorphism (static)
- Run-time polymorphism (dynamic)

14. Base Class

15. Advantages:

- Easy to maintain and modify code
- Provides modular structure for programs

16. It is the ability to process object differently depending on their data type or class.

- **Static Polymorphism:** it refers to the ability of the compiler to relate or bind the function call with function definition during compilation itself. There are two methods for static polymers function overloading and operator overloading
- **Dynamic Polymorphism:** It refers to binding the function definition with function call during runtime.