

Virtual Functions Exercise

1. What is function overloading?

- (a) Same function name with different parameters
- (b) Same function in different classes
- (c) One function only

→(a) Same function name with different parameters

2. Function overloading occurs in:

- (a) Same class
- (b) Different classes
- (c) Base class only

→(a) Same class

3. What is function overriding?

- (a) Using many functions
- (b) Same function name in base and derived class
- (c) Calling a function

→(b) Same function name in base and derived class

4. Function overriding occurs in:

- (a) Same class
- (b) Different classes
- (c) Same parameters only

→(b) Different classes

5. Which keyword is used to achieve runtime polymorphism?

- (a) static
- (b) virtual
- (c) inline

→(b) virtual

6. Write a C++ program to show function overloading using addition of two and three numbers.

→

```
#include <iostream>
using namespace std;

class Add {
public:
    int sum(int a, int b) {
        return a + b;
    }

    int sum(int a, int b, int c) {
        return a + b + c;
    }
};

int main() {
    Add obj;
    cout << obj.sum(2, 3) << endl;
    cout << obj.sum(2, 3, 4);
    return 0;
}
```

Output-

Output

5
9

7. Write a simple program to demonstrate function overriding using virtual function.

→

```
#include <iostream>
using namespace std;

class Base {
public:
    virtual void show() {
        cout << "Base class function";
    }
};

class Derived : public Base {
public:
    void show() {
        cout << "Derived class function";
    }
};

int main() {
    Base* b;
    Derived d;
    b = &d;
    b->show();
    return 0;
}
```

Output-

Output

Derived class function

8. Fill in the blank:

To call derived class function using base pointer, the function must be virtual.

9. Predict the output when a base class pointer points to a derived class object.

- If function is virtual → derived class function runs
 - If not virtual → base class function runs
-

10. True or False:

Function overloading depends on return type only.

- False
-

11. True or False:

Virtual functions support runtime polymorphism.

- True
-

12. Choose the correct answer:

Which is needed for function overriding?

- (a) Inheritance
- (b) Loops
- (c) Arrays

- (a) Inheritance
-

13. Fill in the blank:

Function overloading is also known as compile time polymorphism.

14. Fill in the blank:

Function overriding is also known as runtime polymorphism.

15. Write a simple program to overload a function named display().

→

```
● ● ●

#include <iostream>
using namespace std;

class Demo {
public:
    void display() {
        cout << "No argument" << endl;
    }

    void display(int x) {
        cout << "Value: " << x;
    }
};

int main() {
    Demo d;
    d.display();
    d.display(10);
    return 0;
}
```

Output-

Output

No argument

Value: 10

16. Write a base and derived class to show function overriding.

→

```
#include <iostream>
using namespace std;

class Base {
public:
    virtual void show( ) {
        cout << "Base class";
    }
};

class Derived : public Base {
public:
    void show( ) {
        cout << "Derived class";
    }
};

int main() {
    Base* b;
    Derived d;
    b = &d;
    b->show();
    return 0;
}
```

Output-

Output

Derived class

17. Identify the error:

A base class pointer calls a derived class function without virtual keyword.

→ Error: Function will not override properly

18. Very short answer:

Why do we use virtual functions?

→ To call the correct function at runtime

—X—X—X—X—X—