

**Q1** Word polymorphism means having \_\_\_\_\_.

**Q2** If derived class defines same function as defined in its base class, it is known as \_\_\_\_\_.

- a) Function overloading
- c) dynamic polymorphism

- b) friend function
- d) none of above

**Q3** In C++, when is dynamic polymorphism achieved?

- a) During compile-time
- b) During runtime
- c) During both compile-time and runtime
- d) Dynamic polymorphism is not possible in C++

**Q4** Polymorphism is achieved through \_\_\_\_\_.

- a) Inheritance
- b) Poly programming
- c) Encapsulation
- d) Overloading

**Q5** What is an abstract class in C++?

- a) A class that cannot be instantiated
- b) A class that contains only pure virtual functions
- c) A class that contains at least one virtual function
- d) A class that is defined with the keyword "abstract"

**Q6** What is the key difference between function overloading and function overriding in C++ polymorphism?

- a) Overloading involves functions with the same name but different parameter lists, while overriding involves functions with the same name and parameter lists in a base and derived class.
- (b) Overloading is static binding (resolved at compile time), while overriding uses runtime binding (virtual functions).
- (c) Overloading can occur within the same class, while overriding requires inheritance.
- (d) Both overloading and overriding are used for dynamic memory allocation.

**Q7** \_\_\_\_\_ binding means that an object is bound to its function call at compile time.

- a) Late
- b) Dynamic

- b) Static
- d) Fixed

**Q8 What will be the output of following code snippet**

(3)

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

class A {
public:
    A(){
        cout<<"Hello from A"\;
    }
    virtual void display(){
        cout<<"Displaying A"\;
    }
};

class B :public A{
public:
    B(){
        cout<<"Hello from B"\;
    }
    void display(){
        cout<<"Displaying B"\;
    }
};

int main() {
    A *a = new B();
    a->display();
    delete a;

    return 0;
}
```

**Output:**