**Section: B** # Quiz 1 **Roll number:** [redacted]

**Q1. What is the main purpose of polymorphism in OOP?** **[1]**

*a) To increase code redundancy*
[✓ *b) To allow different classes to be treated as instances of the same class*]
*c) To prevent objects from being modified*
*d) To ensure type safety*

**Q2: In which of the following scenarios would you use inheritance?** **[1]**

*a) Create multiple objects of the same class*
*b) Share code between different classes that have common functionality*
*c) Restrict access to certain parts of your code*
*d) Execute a block of code repeatedly*

**Q3: What is the key difference between method overloading and method overriding?** **[1]**

*a) Overloading allows different methods to have the same name but different signatures; overriding allows a subclass method to replace a superclass method*
*b) Overloading is done in the same class; overriding is done in different classes.*
*c) Overloading occurs at runtime; overriding occurs at compile-time.*
*d) Overloading is more flexible than overriding.*

**Q4: Consider the following C++ code:**

DSA

```
class Base {
public:
        virtual void show() {
        cout << "Base class" << endl;
        }
};
class Derived : public Base {
public:
        void show() override {
        cout << "Derived class" << endl;
        }
};
int main() {
        Base* b = new Derived();
        b->show();
        return 0;
}
```

Base class

**What will be the output of the above code?** **[1]**

*a) Base class*
*b) Runtime error*
*c) Compilation error*
*d) Derived class*

**Q5: Which of the following is NOT a benefit of encapsulation in OOP?** **[1]**

*a) Improved code maintenance*
[✓ *b) Increased code flexibility*]
*c) Direct access to private data members*
*d) Enhanced data security*

Section: B

# Quiz 1

Roll number: BSSE23097 Section DSA

***Q6: You are given a base class 'Shape' with a virtual function 'area()'. Derive two classes, 'Rectangle' and 'Circle', from the 'Shape' class. Override the 'area()' function in both derived classes to calculate and return the area of the respective shapes.***

***Create objects of 'Rectangle' and 'Circle', store them in 'Shape' pointers, and display their areas using polymorphism.***

```
#include <iostream>
using namespace std;
class Shape {
public:
    virtual double area() const = 0;
};

int main() {
    // Your Implementation
    return 0;
}
```

cla