```cpp
class Animal{

public:

    ~Animal();

    virtual void eat(Animal *) const;

    virtual void sleep(std::ostream) const;

    virtual void speak(std::ostream) const = 0;


    std::string getColor() const;

    int getWeight() const;


    void setColor(const std::string);

    void setWeight(int);

protected:

    Animal(std::string, int);

private:

    std::string color_;

    int weight_;

};

// 1. Implement Animal ctor that initializes all data members

//

//

inline Animal::~Animal() { }

inline std::string Animal::getColor() const { return color_; }

inline int Animal::getWeight() const { return weight_; }

inline void Animal::setColor(const std::string color) { color_ =
color; }

inline void Animal::setWeight(int weight) { weight_ = weight; }
```

```
class Mammal: public Animal{
public:
    ~Mammal();
    virtual void sleep(std::ostream);
    virtual void speak(std::ostream) const = 0;
    std::string getName() const;
    int getAge() const;
    void setName(const std::string);
    void setAge(int);
protected:
    Mammal();

    //2a. Declare ctor for Mammal that initializes all data members
    //
private:
    std::string name_;
    int age_;
};

//2b.  Implement ctor for Mammal that initializes all data members
//

inline Mammal::~Mammal() { }
inline void Mammal::sleep(std::ostream os) {
    os << "zzzZ zzZz zzZ zz" << std::endl;
}
inline std::string Mammal::getName() const { return name_; }
inline int Mammal::getAge() const { return age_; }
inline void Mammal::setName(const std::string name) { name_ = name; }
inline void Mammal::setAge(int age) { age_ = age; }
```

```
class Dog: public Mammal {

public:

    // 4a. Declare ctor for Dog that initializes all derived data
members

    //


    ~Dog();


    // 6a. Declare a copy assignment operator for Dog

    //

    // 8. Overload << operator and >> operator for Dog

    //

    //9b. Declare all required methods for Dog here and implement them
    below

    //

};


//4b. Implement ctor for Dog that initializes all derived data members

//


//6b. Implement the copy assignment operator for Dog

//


//9b. Implement required methods here

//


Dog::~Dog() { }
```

```
//Cat.h
class Cat: public Mammal {
public:
    //3a. Declare ctor for Cat that initializes all derived data
members
    //

    //5a. Define a copy ctor for Cat
    //

    Cat(Cat& c);
    ~Cat();

    //7. Declare a default assignment operator for Cat

    //9a. Declare all required methods for Cat here and implement them
in the header file

    std::string getBreed() const;

private:
    std::string breed_;
};
```

```
//Cat.cpp

//3b. Implement ctor for Cat that initializes all derived data members



//5b. Implement a copy ctor for Cat


//9a. Implement required methods here


Cat::~Cat() { }


std::string Cat::getBreed() const { return breed_; }
```

Using the code given in the previous pages, fill in the missing code.

1. Implement Animal ctor

2a. Declare a ctor for Mammal that initializes all data members
2b. Implement the ctor for Mammal that initializes all data members

3a. Declare ctor for Cat that initializes all data members
3b. Implement ctor for Cat that initializes all data members

4a. Declare ctor for Dog that initializes all data members
4b. Implement ctor for Dog that initializes all data members

5a. Declare a copy ctor for Cat that initializes all data members
5b. Implement a copy ctor for Cat that initializes all data members

6a. Declare a copy assignment operator for Dog
6b. Implement a copy assignment operator for Dog

7. Declare a default assignment operator for Cat

8. Overload << and >> operators for Dog.

9a. Implement REQUIRED methods for Cat
9b. Implement REQUIRED methods for Dog

Short Answers Questions:

1. What does the explicit keyword in front of a method or ctor do?

2. What does friend in front of a method or operator do?

3. What does the default keyword at the end of a ctor, dtor, or operator overload do?

4. What is the difference between a virtual method and a pure virtual method?

5. True/False, explicit and virtual are required to be stated in both the declaration and implementation.