

1. What is wrong with the following code and how would you fix it?

```
#ifndef PROJECTILE_H
#define PROJECTILE_H
class Projectile {
private:
    double position;
    double velocity;
public:
    Projectile(double position, double velocity);
    virtual ~Projectile();
    double getPosition() const;
    double getVelocity() const;
} // end of Projecile class
#endif
```

Answer:

There was a semi colon missing at the end.

```
#ifndef PROJECTILE_H
#define PROJECTILE_H
class Projectile {
private:
    double position;
    double velocity;
public:
    Projectile(double position, double velocity);
    virtual ~Projectile();
    double getPosition() const;
    double getVelocity() const;
};
#endif
```

2. The following is the definition of the constructor for the Projectile class above, but there are three things wrong with it. What are they and how would you fix them?

```
Projectile(int position, int velocity) {
    this.position = position;
    this.velocity = velocity;
} // end of constructor
```

Answer:

```
Projectile::Projectile(double position, double velocity) {  
    this ->position = position;  
    this ->velocity = velocity;  
};
```

```
Projectile::~Projectile~();
```

```
// end of constructor
```

3. Describe each of the following methods

- (a) `int* method(int* arg);`**
- (b) `const int* method(int* arg);`**
- (c) `const int* const method(int* arg);`**
- (d) `const int* const method(const int* arg);`**
- (e) `const int* const method(const int* arg) const;`**

Answer:

(A) *int that is a pointer to a method that contains an int the points to an argument.*

(B) *a method containing a int that is a pointer to an argument that points to a const int.*

(C) *const method that is a pointer to a const int that contains an int that is a pointer to an argument.*

(D) *const method that contains a const int that points to a argument that points to a const int.*

(E) *a const with a const in pointing to a const method that contains a const int pointing to a argument.*

4. In what ways are C++ strings better than C strings? In what ways

are C strings better than C++ strings?

Answer:

C++ strings are better than C strings because they are more convenient, reliable, and more familiar with java users. C strings are better than C++ strings because it can be used to write performance-critical code.

5. What is the difference between a pointer and a reference?

Answer:

.A pointer returns multiple values from functions, references memory locations, and refers to sub-arrays. A reference automatically dereferences, has to be assigned when declared, programmer/user can not change what it points to and it can not be a NULL .

6. What is a destructor for?

Answer:

Destructors are used to delete heap data created by an object.