#include <iostream.h>

class Ball {
private:
    const float pi;
    int radius;

public:
    Ball(int r=1) {
        radius = r;
    }
    void set_radius(int radius);
    const Ball& operator=(const Ball& b);
    static int count;
    virtual void print() {
        cout << radius << endl;
    }
};

int Ball::count = 0;

class BuckyBall: public Ball {
    private:
        int color;

    public:
        BuckyBall(int radius, int c) {
            color = c;
        }
        void print() {
            cout << color << endl;
        }
};
Question 1. Show how you would initialize the member $pi$ in class $Ball$.

Answer:

Question 2. Write the copy constructor for class $Ball$.

Answer:

Question 3. Show how you would overload the $+=$ operator, so that the following code increments the radius of $b$ by 2.

```
Ball b;
b += 2;
```

Answer:
Question 4. Complete the definition of the member function `set_radius()`.

```cpp
void Ball::set_radius(int radius) {
    // Answer:
}
```

Question 5. What should the = operator return so that the code

```cpp
Ball a, b(2), c(3);
a = b = c;
```

behaves as expected? Explain your answer.

```cpp
const Ball& Ball::operator=(const Ball& b) {
    radius = b.radius;
    // Answer:
}
```

Question 6. Draw a clear diagram to illustrate the memory allocated by the following code. Label all variables on your diagram.

```cpp
Ball b;
Ball *p;
Ball **pp;

pp = new Ball*[2];
pp[0] = new Ball[2];
pp[1] = &b;
Ball& c = pp[0][1];
```

Answer:
Question 7. How you would release the memory allocated in Question 6?

Answer:

Question 8. What will be the output from the following program?

```cpp
int count = 5;

void draw(Ball *p, int n) {
    static int count = n;
    cout << count << endl;
}

void main() {
    const int count = 2;
    Ball b[count];
    draw(b,7);
    draw(b,8);
    cout << b[1].count << count << ::count << Ball::count << endl;
}
```

Answer:
**Question 9.** Show how you would modify the *BuckyBall* constructor so that it correctly initializes the *Ball* part of a *BuckyBall* object.

**Answer:**

**Question 10.** What statements would you use to print out

(i) The color of object *a*?
(ii) The color of object *b*?
(iii) The radius of object *b*?
(iv) The radius of object *c*?

*BuckyBall* *a(1,2)*;
*BuckyBall& b = a*;
*BuckyBall& c = a*;

**Answer:**
Question 11. What is a protected member? Give examples of how such a member can and cannot be used.

Answer:

Question 12. Give the definitions of the destructors for the Ball and BuckyBall classes.

Answer: