MINOR ASSIGNMENT-04

Game Programming with C++ (CSE 3545)

Publish on: 12-04-2025 **Submission on:** 18-04-2025
Course Outcome: CO_3 **Program Outcome:** PO_3 **Learning Level:** L_4

Problem Statement:

Experiment with objects by coding various classes for Pong game and to explore the benefits of Object-Oriented Programming(OOP) paradigm in designing the game.

Learning Objectives:

Students will be able to learn and use OOP to get started with the Pong game project by coding own classes.

Answer the followings:

1. Create a code snippet to declare two private members of the type `Vector2f` and `RectangleShape` for the class **HypoBat** with appropriate headers.

Code Snippet

```
#pragma once
#include <SFML/Graphics.hpp>

class HypoBat {
private:
    sf::Vector2f m_Position;
    sf::RectangleShape m_Shape;
};
```

2. Fill out the places marked with the symbol, **?**, in the following code snippet.

```
Font ?;
?.loadFromFile("sample.ttf");
Text  ?;
?.setFont(?);
?.?(Color::White);
? . setCharacterSize(75);
```

Code Snippet

```
Font font;
font.loadFromFile("sample.ttf");
Text text;
text.setFont(font);
text.setFillColor(Color::White);
text.setCharacterSize(75);
```

3. Assume that **MyBat** class has four `int` type data members and two member functions, **setData()** and **getData()** with return types `void`. Write the code snippet to declare the said class.

Code Snippet

```
class MyBat {
private:
    int x, y, width, height;

public:
    void setData();
    void getData();
};
```

4. Write the public member functions definition outside of the class for question-3. The function **setData()** to initialize the data members and **getData()** to display the data members.

Code Snippet

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

void MyBat::setData() {
    x = 100;
    y = 200;
    width = 50;
    height = 10;
}

void MyBat::getData() {
    cout << "X: " << x << ", Y: " << y << endl;
    cout << "Width: " << width << ", Height: " << height << endl;
}
```

5. As encapsulation in action, the class members variables cannot be accessed directly from `main`. So Write the code snippet to access the members variables from `main` indirectly by the code of the class using an object of the class **MyBat**.

Code Snippet

```
int main() {
    MyBat bat;
    bat.setData();
    bat.getData();
    return 0;
}
```

6. The above declared class of yours provide two functions that are `public` and will be usable with an object (*i.e. an instance of the class*) of the `MyBat` type. Write the code snippet to create FOUR instances of that class and access the public functions by one of them.

Code Snippet

```
int main() {
    MyBat bat1, bat2, bat3, bat4;

    bat1.setData();
    bat1.getData(); // Accessing functions using bat1

    return 0;
}
```

7. Write a program to design a class with private data members and public functions as necessary to draw a rectangle shape of size (10, 10) over a window of resolution 1920 & 1080 respectively.

Code Snippet

```
include <SFML/Graphics.hpp>

class MyRectangle {
private:
    sf::RectangleShape rectangle;

public:
    MyRectangle() {
        rectangle.setSize(sf::Vector2f(10, 10));
        rectangle.setPosition(100, 100); // example position
        rectangle.setFillColor(sf::Color::Green);
    }

    sf::RectangleShape getShape() {
        return rectangle;
    }
};

int main() {
    sf::RenderWindow window(sf::VideoMode(1920, 1080), "Draw
Rectangle");

    MyRectangle myRect;

    while (window.isOpen()) {
        sf::Event event;
        while (window.pollEvent(event)) {
            if (event.type == sf::Event::Closed)
                window.close();
        }

        window.clear();
        window.draw(myRect.getShape());
        window.display();
    }

    return 0;
}
```

8. Design a **SelfBat** class with a parameterize constructor to takes two `float` parameters. Write a program to create a **bat** of size **100**$\times$**5**. The constructor receives two values that represent the position of the bat on the screen.

Code Snippet

```
#include <SFML/Graphics.hpp>

class SelfBat {
private:
    sf::RectangleShape bat;

public:
    SelfBat(float startX, float startY) {
        bat.setSize(sf::Vector2f(100, 5));
        bat.setPosition(startX, startY);
        bat.setFillColor(sf::Color::Blue);
    }

    sf::RectangleShape getShape() {
        return bat;
    }
};

int main() {
    sf::RenderWindow window(sf::VideoMode(1920,
1080), "SelfBat Example");

    SelfBat bat(500, 500);

    while (window.isOpen()) {
        sf::Event event;
        while (window.pollEvent(event)) {
            if (event.type == sf::Event::Closed)
                window.close();
        }

        window.clear();
        window.draw(bat.getShape());
        window.display();
    }

    return 0;
}
```

10. Write the **`update(Time dt)`** public member function definition of our designed **PONG!!!** game with appropriate member variables.

Code Snippet

```
void Bat::update(sf::Time dt) {
    if (m_MovingLeft)
        m_Position.x -= m_Speed * dt.asSeconds();
    if (m_MovingRight)
        m_Position.x += m_Speed * dt.asSeconds();

    m_Shape.setPosition(m_Position);
}
```

11. We have `moveLeft`, `moveRight`, `stopLeft` and `stopRight` functions in our **Bat** class of the **PONG!!!** game for controlling the direction the bat will be in motion. Additionally, we found the bat is getting out of the window scene. Now re-write the required functions so that the bat would not move out of the window (i.e. always visible on the window).

Code Snippet

```
void Bat::moveLeft() {
    m_MovingLeft = true;
}

void Bat::moveRight() {
    m_MovingRight = true;
}

void Bat::stopLeft() {
    m_MovingLeft = false;
}

void Bat::stopRight() {
    m_MovingRight = false;
}

void Bat::update(sf::Time dt) {
    if (m_MovingLeft && m_Position.x > 0)
        m_Position.x -= m_Speed * dt.asSeconds();
    if (m_MovingRight && m_Position.x + m_Shape.getSize().x < 1920)
        m_Position.x += m_Speed * dt.asSeconds();

    m_Shape.setPosition(m_Position);
}
```

12. Write a event poll loop to display a message, `A Key Pressed`, on the standard stream(i.e. monitor), when an event **KeyPressed** would be happened. Further add few lines of code to detect whether the key **W** is pressed or any other key.

Code Snippet

```
while (window.pollEvent(event)) {
    if (event.type == sf::Event::KeyPressed) {
        std::cout << "A Key Pressed" << std::endl;
        if (event.key.code == sf::Keyboard::W) {
            std::cout << "W Key Pressed" << std::endl;
        } else {
            std::cout << "Another Key Pressed" << std::endl;
        }
    }
}
```

13. State the code snippet to handle the ball hitting the top.

Code Snippet

```
if (ball.getPosition().top < 0) {
    ball.reboundBatOrTop();
}
```

```
void Ball::reboundBatOrTop(){
    m_directionY=-m_directionY;
}
```

14. State the code snippet to handle the ball hitting the buttom.

Code Snippet

```
if (ball.getPosition().top > window.getSize().y) {
    ball.hitBottom();
}
```

```
void Ball::reboundBottom(){
    m_position.y=0;
    m_position.x=400;
    m_directionY=-m_directionY;
}
```

15. State the code snippet to handle the ball hitting the sides.

Code Snippet

```
if (ball.getPosition().left < 0 || ball.getPosition().left
+ ball.getPosition().width > window.getSize().x) {
    ball.reboundSides();
}
```

```
void Ball::reboundSides(){
    m_directionX=-m_directionX;
}
```

16. State the code snippet to to determine whether the ball has hit the bat (dynamic collision detection).

Code Snippet

```
if (ball.getPosition().intersects(bat.getPosition())) {
    ball.reboundBatOrTop();
}
```

17. Consider the following C++ code snippet;

```
class CSE{
 public:
   int x, y;
   void set(int x1, int y1){
      x=x1;y=y1;
  }
  void get(){
      cout<<x<<" "<<y<<endl;
  }
};
int main(){
        CSE a;a.set(10,20);
        a.get();
        return 0;
}
```

Output

Output:
10 20

18. Consider the following C++ code snippet;

```
class CSE{
 public:
   int x, y;
   CSE(int x1, int y1){
      x=x1;y=y1;
  }
  void get(){
     cout<<x<<" "<<y<<endl;
  }
};
int main(){
    CSE(100,200).get();
    CSE A(50,60);
    cout<<A.x<<" "<<A.y<<endl;
    return 0;
}
```

Output

Output:
100 200
50 60

19. Consider the following C++ code snippet;

```
 class Box{
        public :
        double length;
        double breadth;
        double height;
 };
int main(){
Box Box1;
double volume;
Box1.height = 5;
Box1.length = 6;
Box1.breadth = 7.1;
volume = Box1.height * Box1.length * Box1.
   breadth;
cout << "Volume of Box1 : " << volume <<endl;
return 0;
}
```

Select the Correct one

- ❑ 210
- [✓ 213]
- ❑ 215
- ❑ 217