UNIVERSITY OF CALIFORNIA, MERCED

BERKELEY · DAVIS · IRVINE · LOS ANGELES · MERCED · RIVERSIDE · SAN DIEGO · SAN FRANCISCO

SANTA BARBARA · SANTA CRUZ

CSE 24: Advanced Programming

Midterm Examination

Spring 2024

First Name _____ **Last Name** _____

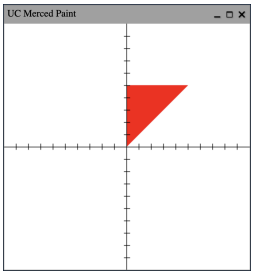
UC Merced Email _____

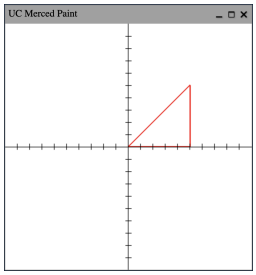
Instructions

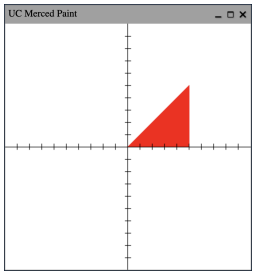
- Write your first and last name as well as your official UC Merced on the Scantron and Exam Booklet.
- You must turn in both your Exam Booklet with your Scantron.
- This is a closed book exam. No notes and/or electronic devices may be used.
- Answer every question on the Scantron. You can use any blank part of the exam as scratch paper.
- You have 1 hours and 15 minutes to complete this exam.
- If you are unsure of anything, please ask.

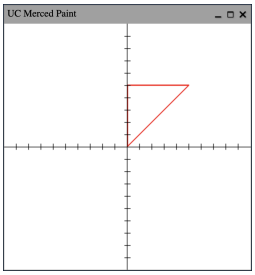
1. What is the output of the following code snippet?

```
glColor3f(1.0f, 0.0f, 0.0f);

glBegin(GL_POLYGON);
    glVertex2f(0.0f, 0.0f);
    glVertex2f(0.5f, 0.0f);
    glVertex2f(0.5f, 0.5f);
glEnd();
```

A. 

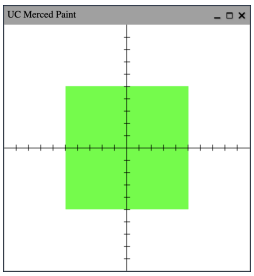
B. 

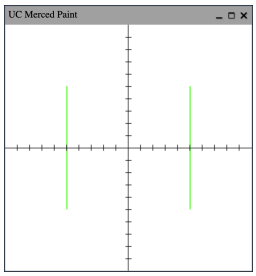
C. 

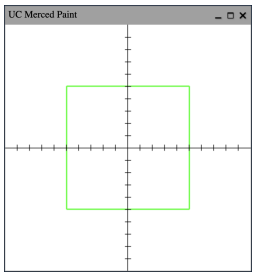
D. 

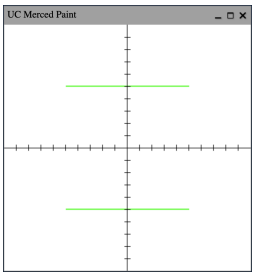
2. What is the output of the following code snippet?

```
glLineWidth(2.0f);
glColor3f(0.0f, 1.0f, 0.0f);

glBegin(GL_LINES);
    glVertex2f(-0.5f, 0.5f);
    glVertex2f(-0.5f, -0.5f);
    glVertex2f(0.5f, -0.5f);
    glVertex2f(0.5f, 0.5f);
glEnd();
```

A. 

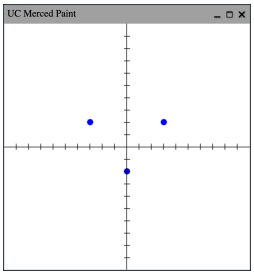
B. 

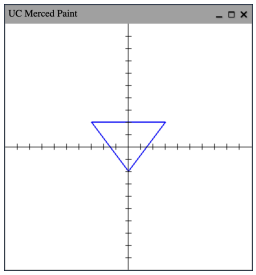
C. 

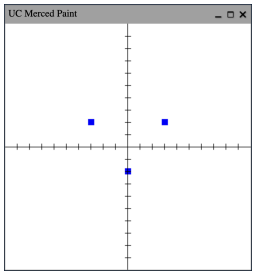
D. 

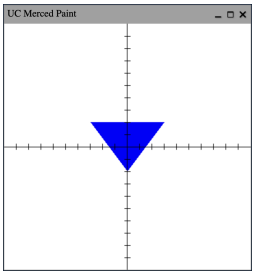
3. What is the output of the following code snippet?

```
glPointSize(10.0f);
glColor3f(0.0f, 0.0f, 1.0f);

glBegin(GL_POINTS);
    glVertex2f(-0.3f, 0.2f);
    glVertex2f(0.0f, -0.2f);
    glVertex2f(0.3f, 0.2f);
glEnd();
```

A. 

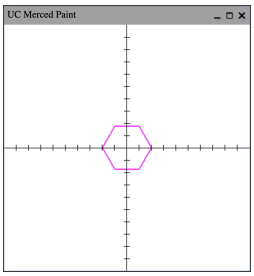
B. 

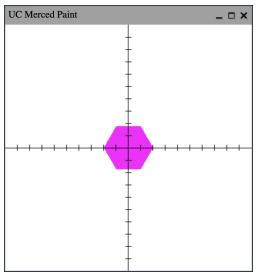
C. 

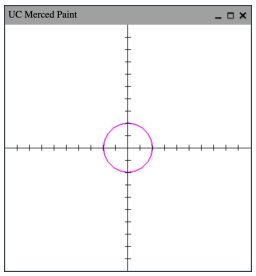
D. 

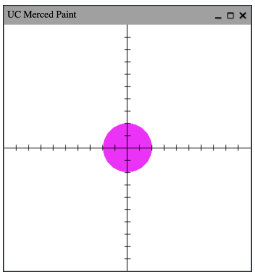
4. What is the output of the following code snippet?

```
float inc = (2 * M_PI) / 6;
float r = 0.2f;

glColor3f(1.0f, 0.0f, 1.0f);
glBegin(GL_POLYGON);
for (float theta = 0; theta < 2 * M_PI; theta += inc) {
    glVertex2f(r * cos(theta), r * sin(theta));
}
glEnd();
```

A. 

B. 

C. 

D. 

5. What color is represented by `glColor3f(0.0f, 0.0f, 0.0f)`?

   A. Black
   B. White
   C. Light Gray
   D. None of the above.

The following 2 structs (`Color` and `Square`) are related to questions 6-10. The drawing in Figure 1 represents how a `Square` object should be rendered from its data members.

```
struct Color {
  float r, g, b;

  Color() {
    r = 1.0f;
    g = 0.0f;
    b = 0.0f;
  }

  Color (float r, float g, float b) {
    this->r = r;
    this->g = g;
    this->b = b;
  }
};
```

```
struct Square {
  float x;
  float y;
  float size;
  Color color;
};
```

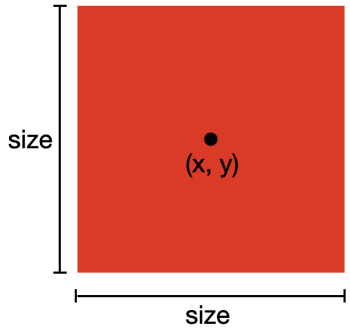


Figure 1: Square of `size` centered at coordinates `x` and `y`

6. Which of the following is an appropriate default constructor for the `Square` struct?

A.
```
square() {
  x = 0.0f;
  y = 0.0f;
  size = 0.2f;
}
```

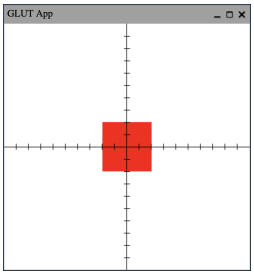
B.
```
Square() {
  x = 0.0f;
  y = 0.0f;
  size = 0.2f;
}
```

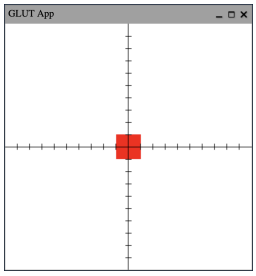
C.
```
Square() {
  x = 0.0f;
  y = 0.0f;
}
```

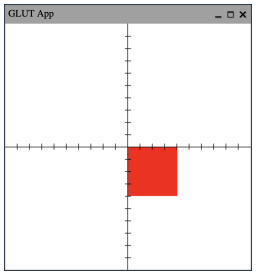
D.
```
Square() {
  x = 0.0f;
  y = 0.0f;
  size = 0.2f;
  color = Color(1.0f, 0.0f, 0.0f);
}
```

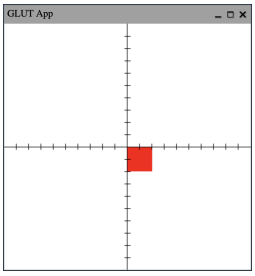
E. Both B and D.

7. What is the output of drawing a `Square` using the default constructor from the previous question, where the `size` is `0.2` and `(x, y)` coordinates are `(0.0, 0.0)`?

A. 

B. 

C. 

D. 

8. Which of the following is an appropriate parametrized constructor for the `Square` struct?

A.
```
Square(float x, float y, float size) {
    this->x = x;
    this->y = y;
    this->size = size;
}
```

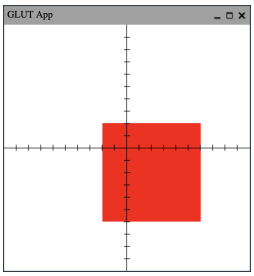
B.
```
square(float size) {
    x = 0.0f;
    y = 0.0f;
    this->size = size;
}
```

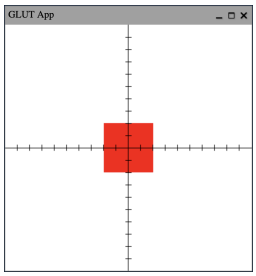
C.
```
Square(float x, float y) {
    this->x = x;
    this->y = y;
    size = 0.2f;
}
```

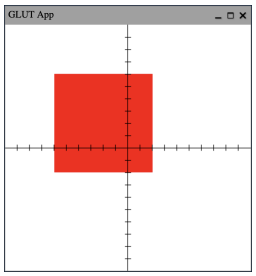
D. All of the above

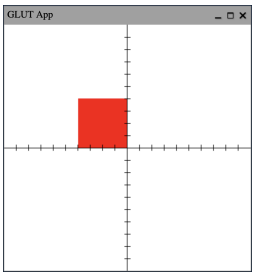
E. Both A and C

9. What is the output of drawing a `Square` whose `size` is `0.4` and `(x, y)` coordinates are `(-0.2, 0.2)`?

A. 

B. 

C. 

D. 

10. Which of the following is the correct `draw` method for the `Square` struct?

A.
```
void draw() {
    glColor3f(color.r, color.g, color.b);

    glBegin(GL_POLYGON);
        glVertex2f(x - size, y + size);
        glVertex2f(x + size, y + size);
        glVertex2f(x + size, y - size);
        glVertex2f(x - size, y - size);
    glEnd();
}
```

B.
```
void draw() {
    glColor3f(color.r, color.g, color.b);

    glBegin(GL_POLYGON);
        glVertex2f(x - size/2, y + size/2);
        glVertex2f(x + size/2, y + size/2);
        glVertex2f(x + size/2, y - size/2);
        glVertex2f(x - size/2, y - size/2);
    glEnd();
}
```

C.
```
void draw() {
    glColor3f(color.r, color.g, color.b);

    glBegin(GL_POLYGON);
      glVertex2f(x, y);
      glVertex2f(x + size/2, y);
      glVertex2f(x + size/2, y - size/2);
      glVertex2f(x, y - size/2);
    glEnd();
}
```

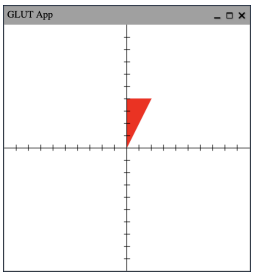
D.
```
void draw() {
    glColor3f(color.r, color.g, color.b);

    glBegin(GL_POLYGON);
      glVertex2f(x, y);
      glVertex2f(x + size, y);
      glVertex2f(x + size, y - size);
      glVertex2f(x, y - size);
    glEnd();
}
```

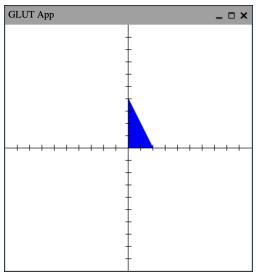
The following 2 structs (`Color` and `Triangle`) are related to questions 11-17.

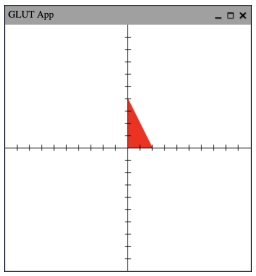
```
struct Color {
  float r, g, b;

  Color() {
    r = 0.0f;
    g = 0.0f;
    b = 1.0f;
  }

  Color (float r, float g, float b) {
    this->r = r;
    this->g = g;
    this->b = b;
  }
};

struct Triangle {
  float x, y, b, h;
  Color color;
  bool selected;

  Triangle() {
      x = 0.0f;
      y = 0.0f;
      b = 0.2f;
      h = 0.4f;
      selected = true;
  }

  Triangle(float x, float y, float b, float h) {
      this->x = x;
      this->y = y;
      this->b = b;
      this->h = h;
      selected = false;
  }

  void draw() {
      if (selected) {
          glColor3f(1.0f, 0.0f, 0.0f);
      } else {
          glColor3f(color.r, color.g, color.b);
      }
      glBegin(GL_POLYGON);
          glVertex2f(x, y);
          glVertex2f(x, y + h);
          glVertex2f(x + b, y);
      glEnd();
  }
};
```

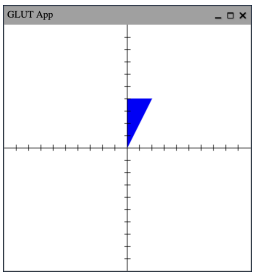
11. What is the output of the following code snippet?

```
Triangle t;
t.draw();
```

A. 

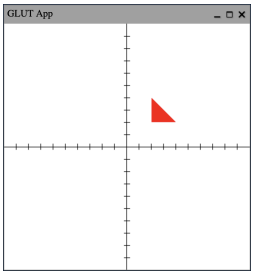
B. 

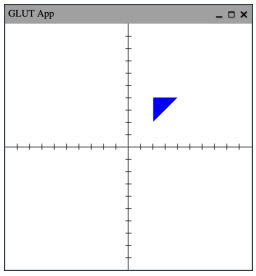
C. 

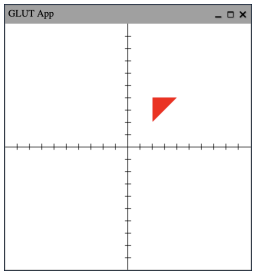
D. 

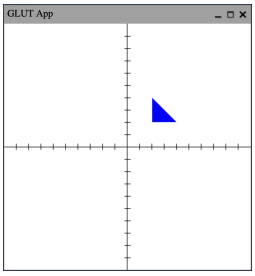
12. What is the output of the following code snippet?

```
Triangle t(0.2f, 0.2f, 0.2f, 0.2f);
t.draw();
```

A. 

B. 

C. 

D. 

13. Which of the following statements is **NOT** true?
    A. Question 11 uses the default constructor.
    B. Question 12 uses the parametrized constructor.
    C. There can only be one default constructor.
    D. There can only be one parametrized constructor.
    E. All of the statements are true.

14. Which of the following statements best describes the `(x, y)` coordinates of the `Triangle` struct?
    A. The coordinates `(x, y)` represent the bottom left corner of the `Triangle`.
    B. The coordinates `(x, y)` represent the top left corner of the `Triangle`.
    C. The coordinates `(x, y)` represent the bottom right corner of the `Triangle`.
    D. None of the above.

15. How many data members does the `Triangle` struct have?
    A. 4
    B. 5
    C. 6
    D. 7

16. Which of the following is a valid way to update the `selected` member of the `Triangle t`?

    A. `t[selected] = false;`

    B. `t.selected = false;`

    C. `t->selected = false;`

    D. All of the above

    E. Both B and C

17. Which of the following declares an array of `Triangles`?

    A. `Triangles triangle[100];`

    B. `Triangle triangles[100];`

    C. `Triangle[100] triangles;`

    D. All of the above

    E. Both A and B

18. What is a C++ struct?

    A. A function that returns multiple values

    B. A user-defined data type that groups related data together

    C. A reserved keyword for declaring variables

    D. A type of loop used for iterating over arrays

19. What is encapsulation in the context of C++ structs?

    A. Combining data and functions into a single unit

    B. Protecting data from unauthorized access

    C. Hiding the implementation details of a struct

    D. Grouping related data together

20. What is abstraction in the context of C++ programming?

    A. Creating objects from classes

    B. Hiding complex implementation details and showing only essential features

    C. Inheriting properties and behaviors from a base class

    D. Defining multiple functions with the same name but different parameters