**1.** Why don’t we “just” use a commercial or open-source graphics library directly?

Because graphic libraries contain a lot of different complex facilities, we should use our interface library in order not to get confused by this variety.

**2.** About how many classes from our graphics interface library do you need to do simple graphics output?

Point, Window (or Simple Window) and all classes that has Shape as a base class (Color, Rectangle, Open\_polyline, Close\_polyline etc.) Total - 19 Classes

**3.** What are the header files needed to use the graphics interface library?

Point.h, Window.h, Simple\_window.h, GUI.h, Graph.h, fltk.h

**4.** What classes define closed shapes?

Close\_polyline, Rectangle, Circle, Ellipse, Polygon

**5.** Why don’t we just use Line for every shape?

If library provides “native” way of representing the shape it is better to use it instead of implementing a function. Such classes like Rectangle, Circle and Ellipse don’t use Line to represent a shape.

**6.** What do the arguments to Point indicate?

The coordinates of point on y and x axes.

**7.** What are the components of Line\_style?

Enumeration Line\_style\_type that holds names of styles (dash, solid etc.), constructor that initializes an object with given style and width 0, constructor that requires style and width and constructor that uses integer representation of style and width 0 to construct an object. Also there are two private members integer style and integer width and two functions that return these values.

**8.** What are the components of Color?

Similar to Line\_style there is an enumeration (Color\_type) that holds all colors provided by FLTK. Also there is an enumeration for transparency levels (invisible = 0, visible = 255), constructors to initialize color, color with transparency, color as integer and only transparency. There is a private member of type char that holds visibility value and member of type Fl\_Color that holds color and functions that return color as integer, visibility as char and set visibility.

**9.** What is RGB?

A color model that uses the mix of three colors (red, green, blue) to represent a broad array of colors.

**10.** What are the differences between two Lines and a Lines containing two lines?

“Lines” are manipulated as one object. That means that all color and style changes will be applied to all lines, while every object of type Line can be changed separately.

**11.** What properties can you set for every Shape?

Starting Point, color, visibility and style.

**12.** How many sides does a Closed\_polyline defined by five Points have?

5 sides

**13.** What do you see if you define a Shape but don’t attach it to a Window?

Nothing, it won’t be displayed

**14.** How does a Rectangle differ from a Polygon with four Points (corners)?

Rectangle has four right angles, while polygon with four points can have angles that are not equal to each other.

**15.** How does a Polygon differ from a Closed\_polyline?

A Polygon is a Closed\_polyline where the lines don’t cross.

**16.** What’s on top: fill or outline?

Outline is on top and some part of fill is closed by it.

**17.** Why didn’t we bother defining a Triangle class (after all, we did define Rectangle)?

Polygon class allows us to define Triangles of various types. As for Rectangles, they are the most common shapes in graphical interfaces and that’s why graphical libraries support them directly.

**18.** How do you move a Shape to another place in a Window?

There is a function move() that takes a number of pixels which has to be added to current x and y coordinates of Shape’s object as an argument.

**19.** How do you label a Shape with a line of text?

There is a class called Text, which can be used to label the Shape. The constructor of this class takes the position and string of text as arguments.

**20.** What properties can you set for a text string in a Text?

Starting Point, Color, Size, Font and Label.

**21.** What is a font and why do we care?

Font is a particular size, width and style of a letter. During the construction of Text’s object, the default FLTK size of letters is checked whether it is less than 14 or not. If yes, the size of letters will be set to 14 pixels. We do this to ensure that the font size that is initialized by default won’t change if the default FLTK size will be changed.

**22.** What is Vector\_ref for and how do we use it?

Vector\_ref is a type of vector that is used to hold both named and unnamed objects. We use it in the same way as the Vector from standard library. To add unnamed objects we need to use operator **new.**

**23.** What is the difference between a Circle and an Ellipse?

Every Circle is the Ellipse, but not every Ellipse is a Circle. The main difference is that the distance from the center to a point on x axis and the distance from the center to a point on y axis in ellipse is not equal, while the distance from center to every point of circle is always the same and is called radius.

**24.** What happens if you try to display an Image given a file name that doesn’t refer to a file containing an image?

The Graphical System will display an object of class Bad Image.

**25.** How do you display part of an image?

To display a part of the image the function set\_mask() has to be used. It requires a top leftmost point of picture and desirable width and height of picture’s part that has to be displayed.