IIT Bombay

Computer Programming

Dr. Deepak B Phatak
Dr. Supratik Chakraborty
Department of Computer Science and Engineering
IIT Bombay

Session: Coordinate based Graphics
Guest Lecturer: Dr. Abhiram Ranade

Quick recap

IIT Bombay

Quick recap

IIT Bombay

Last session: Turtle Graphics facility of Simplecpp.

Quick recap

IIT Bombay

Last session: Turtle Graphics facility of Simplecpp.

This session: Coordinate based graphics.

Quick recap

IIT Bombay

Last session: Turtle Graphics facility of Simplecpp.

This session: Coordinate based graphics.

Chapter 5 of "An introduction to programming through C++", McGraw Hill Education, 2014.

Coordinate based Graphics: Basics

IIT Bombay

Coordinate based Graphics: Basics

IIT Bombay

- Use function `initCanvas` to create graphics window.

Coordinate based Graphics: Basics

IIT Bombay

- Use function `initCanvas` to create graphics window.
- Drawing window, "canvas", has a coordinate frame, with origin at top left corner.
  x axis goes to right. y axis goes down.

Coordinate based Graphics: Basics

IIT Bombay

- Use function `initCanvas` to create graphics window.
- Drawing window, "canvas", has a coordinate frame, with origin at top left corner.
  x axis goes to right. y axis goes down.
- Graphics objects can be created on the canvas.

Coordinate based Graphics: Basics

IIT Bombay

- Use function `initCanvas` to create graphics window.
- Drawing window, "canvas", has a coordinate frame, with origin at top left corner.
  x axis goes to right. y axis goes down.
- Graphics objects can be created on the canvas.
- Graphics objects can be moved, rotated, scaled.

Coordinate based Graphics: Basics

IIT Bombay

- Use function `initCanvas` to create graphics window.
- Drawing window, "canvas", has a coordinate frame, with origin at top left corner.
  x axis goes to right. y axis goes down.
- Graphics objects can be created on the canvas.
- Graphics objects can be moved, rotated, scaled.
- Colour can be changed.

Coordinate based Graphics: Basics

IIT Bombay

- Use function `initCanvas` to create graphics window.
- Drawing window, "canvas", has a coordinate frame, with origin at top left corner.
  x axis goes to right. y axis goes down.
- Graphics objects can be created on the canvas.
- Graphics objects can be moved, rotated, scaled.
- Colour can be changed.
- Graphics objects have pens; lines are drawn when moved.

Coordinate based Graphics: Basics

IIT Bombay

- Use function `initCanvas` to create graphics window.
- Drawing window, "canvas", has a coordinate frame, with origin at top left corner.
  x axis goes to right. y axis goes down.
- Graphics objects can be created on the canvas.
- Graphics objects can be moved, rotated, scaled.
- Colour can be changed.
- Graphics objects have pens; lines are drawn when moved.

Graphics objects are also ordinary C++ objects in memory; member functions modify the memory content as well as the canvas.

Creating graphics objects

IIT Bombay

Creating graphics objects

General form:

```
type name(list of arguments to constructor)
```

IIT Bombay

Creating graphics objects

IIT Bombay

General form:

```
type name(list of arguments to constructor)
```

- `Turtle t1;`

Creating graphics objects

IIT Bombay

General form:

`type name(list of arguments to constructor)`

- `Turtle t1;`
  Creates a turtle named `t1`, at canvas center pointing right.

Creating graphics objects

IIT Bombay

General form:

`type name(list of arguments to constructor)`

- `Turtle t1;`
  Creates a turtle named `t1`, at canvas center pointing right.
- `Circle c1(cx,cy,r);`

Creating graphics objects

IIT Bombay

General form:

`type name(list of arguments to constructor)`

- `Turtle t1;`
  Creates a turtle named `t1`, at canvas center pointing right.
- `Circle c1(cx,cy,r);`
  Creates a circle `c1` centered at `(cx,cy)` of radius `r`.

Creating graphics objects

IIT Bombay

General form:

`type name(list of arguments to constructor)`

- `Turtle t1;`
  Creates a turtle named `t1`, at canvas center pointing right.
- `Circle c1(cx,cy,r);`
  Creates a circle `c1` centered at `(cx,cy)` of radius `r`.
- `Rectangle r(cx, cy, w, h);`

Creating graphics objects

IIT Bombay

General form:

`type name(list of arguments to constructor)`

- `Turtle t1;`
  Creates a turtle named `t1`, at canvas center pointing right.
- `Circle c1(cx,cy,r);`
  Creates a circle `c1` centered at `(cx,cy)` of radius `r`.
- `Rectangle r(cx, cy, w, h);`
  Creates a rectangle `r` centered at `(cx,cy)`, of given width, height.

Creating graphics objects

IIT Bombay

General form:

`type name(list of arguments to constructor)`

- `Turtle t1;`
  Creates a turtle named `t1`, at canvas center pointing right.
- `Circle c1(cx,cy,r);`
  Creates a circle `c1` centered at `(cx,cy)` of radius `r`.
- `Rectangle r(cx, cy, w, h);`
  Creates a rectangle `r` centered at `(cx,cy)`, of given width, height.
- `Line l1(x1,y1,x2,y2);`

Creating graphics objects

IIT Bombay

General form:

`type name(list of arguments to constructor)`

- `Turtle t1;`
  Creates a turtle named `t1`, at canvas center pointing right.
- `Circle c1(cx,cy,r);`
  Creates a circle `c1` centered at `(cx,cy)` of radius `r`.
- `Rectangle r(cx, cy, w, h);`
  Creates a rectangle `r` centered at `(cx,cy)`, of given width, height.
- `Line l1(x1,y1,x2,y2);`
  Creates line named `l1` from `(x1,y1)` to `(x2,y2)`.

Creating graphics objects

IIT Bombay

General form:

`type name(list of arguments to constructor)`

- `Turtle t1;`
  Creates a turtle named `t1`, at canvas center pointing right.
- `Circle c1(cx,cy,r);`
  Creates a circle `c1` centered at `(cx,cy)` of radius `r`.
- `Rectangle r(cx, cy, w, h);`
  Creates a rectangle `r` centered at `(cx,cy)`, of given width, height.
- `Line l1(x1,y1,x2,y2);`
  Creates line named `l1` from `(x1,y1)` to `(x2,y2)`.
- `Text t1(x,y,content);`

Creating graphics objects

IIT Bombay

General form:

`type name(list of arguments to constructor)`

- `Turtle t1;`
  Creates a turtle named `t1`, at canvas center pointing right.
- `Circle c1(cx,cy,r);`
  Creates a circle `c1` centered at `(cx,cy)` of radius `r`.
- `Rectangle r(cx, cy, w, h);`
  Creates a rectangle `r` centered at `(cx,cy)`, of given width, height.
- `Line l1(x1,y1,x2,y2);`
  Creates line named `l1` from `(x1,y1)` to `(x2,y2)`.
- `Text t1(x,y,content);`
  Writes `content` centered at `(x,y)`. The text gets name `t1`.

Creating graphics objects

IIT Bombay

General form:

`type name(list of arguments to constructor)`

- `Turtle t1;`
  Creates a turtle named `t1`, at canvas center pointing right.
- `Circle c1(cx,cy,r);`
  Creates a circle `c1` centered at `(cx,cy)` of radius `r`.
- `Rectangle r(cx, cy, w, h);`
  Creates a rectangle `r` centered at `(cx,cy)`, of given width, height.
- `Line l1(x1,y1,x2,y2);`
  Creates line named `l1` from `(x1,y1)` to `(x2,y2)`.
- `Text t1(x,y,content);`
  Writes `content` centered at `(x,y)`. The text gets name `t1`.
- `Polygon p(cx,cy,coords,n);`

Creating graphics objects

IIT Bombay

General form:

`type name(list of arguments to constructor)`

- `Turtle t1;`
  Creates a turtle named `t1`, at canvas center pointing right.
- `Circle c1(cx,cy,r);`
  Creates a circle `c1` centered at `(cx,cy)` of radius `r`.
- `Rectangle r(cx, cy, w, h);`
  Creates a rectangle `r` centered at `(cx,cy)`, of given width, height.
- `Line l1(x1,y1,x2,y2);`
  Creates line named `l1` from `(x1,y1)` to `(x2,y2)`.
- `Text t1(x,y,content);`
  Writes `content` centered at `(x,y)`. The text gets name `t1`.
- `Polygon p(cx,cy,coords,n);`
  Draws an `n` sided polygon named `p`. Coordinates of vertices are specified in double array `coords`, with 2 rows and `n` columns. Coordinates must be given relative to `(cx,cy)`.

Operations on a graphics object g

IIT Bombay

Operations on a graphics object g

IIT Bombay

Call member functions to manipulate.

Operations on a graphics object g

IIT Bombay

Call member functions to manipulate.

- `g.moveTo(x,y)` : Moves `g` to `(x,y)`.

Operations on a graphics object g

IIT Bombay

Call member functions to manipulate.

- `g.moveTo(x,y)` : Moves `g` to `(x,y)`.
- `g.move(dx,dy)` : Moves `g` by `dx` and `dy` along x and y axes respectively.

Operations on a graphics object g

IIT Bombay

Call member functions to manipulate.

- `g.moveTo(x,y)` : Moves `g` to `(x,y)`.
- `g.move(dx,dy)` : Moves `g` by `dx` and `dy` along `x` and `y` axes respectively.
- `g.scale(factor)` : Scales `g` by given `factor`. Does not apply to `Text`.

Operations on a graphics object g

IIT Bombay

Call member functions to manipulate.

- `g.moveTo(x,y)` : Moves `g` to `(x,y)`.
- `g.move(dx,dy)` : Moves `g` by `dx` and `dy` along `x` and `y` axes respectively.
- `g.scale(factor)` : Scales `g` by given `factor`. Does not apply to `Text`.
- `g.setColor(c)` : Sets the color of `g` to `c`.

Operations on a graphics object g

IIT Bombay

Call member functions to manipulate.

- `g.moveTo(x,y)` : Moves `g` to `(x,y)`.
- `g.move(dx,dy)` : Moves `g` by `dx` and `dy` along `x` and `y` axes respectively.
- `g.scale(factor)` : Scales `g` by given `factor`. Does not apply to `Text`.
- `g.setColor(c)` : Sets the color of `g` to `c`.
  Color `c` can be specified as `COLOR("red")` etc.

Operations on a graphics object g

IIT Bombay

Call member functions to manipulate.

- `g.moveTo(x,y)` : Moves `g` to `(x,y)`.
- `g.move(dx,dy)` : Moves `g` by `dx` and `dy` along `x` and `y` axes respectively.
- `g.scale(factor)` : Scales `g` by given `factor`. Does not apply to `Text`.
- `g.setColor(c)` : Sets the color of `g` to `c`.
  Color `c` can be specified as `COLOR("red")` etc.
  Also as `COLOR(R,G,B)`, where `R,G,B` give intensities of the red, green, blue components between 0 and 255.

Operations on a graphics object g

IIT Bombay

Call member functions to manipulate.

- `g.moveTo(x,y)` : Moves `g` to `(x,y)`.
- `g.move(dx,dy)` : Moves `g` by `dx` and `dy` along `x` and `y` axes respectively.
- `g.scale(factor)` : Scales `g` by given `factor`. Does not apply to `Text`.
- `g.setColor(c)` : Sets the color of `g` to `c`.
  Color `c` can be specified as `COLOR("red")` etc.
  Also as `COLOR(R,G,B)`, where `R,G,B` give intensities of the red, green, blue components between 0 and 255.
- `g.setFill(v)` : If `v` is `true`, then the interior of the object is filled, otherwise the object is drawn as an outline. Applicable only for `Circle, Rectangle, Polygon`

More operations (on object g)

IIT Bombay

More operations (on object g)

IIT Bombay

- `g.hide()` : Hides g.

More operations (on object g)

IIT Bombay

- `g.hide()` : Hides g.
- `g.show()` : Makes g visible, if hidden earlier.

More operations (on object g)

IIT Bombay

- `g.hide()` : Hides g.
- `g.show()` : Makes g visible, if hidden earlier.
- `left, right, forward` are available as member functions.

More operations (on object g)

IIT Bombay

- `g.hide()` : Hides `g`.
- `g.show()` : Makes `g` visible, if hidden earlier.
- `left, right, forward` are available as member functions.
- `g.rotate(A)` : Rotate `g` clockwise. `A` is in radians.

More operations (on object g)

IIT Bombay

- `g.hide()` : Hides g.
- `g.show()` : Makes g visible, if hidden earlier.
- `left, right, forward` are available as member functions.
- `g.rotate(A)` : Rotate g clockwise. `A` is in radians.
- `g.getX(), g.getY(), g.getOrientation(), g.getScale()` : Returns the specified information about g.

More operations (on object g)

IIT Bombay

- `g.hide()` : Hides g.
- `g.show()` : Makes g visible, if hidden earlier.
- `left, right, forward` are available as member functions.
- `g.rotate(A)` : Rotate g clockwise. `A` is in radians.
- `g.getX(), g.getY(), g.getOrientation(), g.getScale()` : Returns the specified information about g.
- g.imprint() : An image of g is drawn at the current position. The image will persist even if g moves.

More operations (on object g)

IIT Bombay

- `g.hide()` : Hides g.
- `g.show()` : Makes g visible, if hidden earlier.
- `left, right, forward` are available as member functions.
- `g.rotate(A)` : Rotate g clockwise. `A` is in radians.
- `g.getX(), g.getY(), g.getOrientation(), g.getScale()` : Returns the specified information about g.
- g.imprint() : An image of g is drawn at the current position. The image will persist even if g moves.
- `g.reset(arg-list)` : Will cause g to be re-initialized using the given `arg-list`, which must have the same form as at creation.

More operations (on object g)

IIT Bombay

- `g.hide()` : Hides g.
- `g.show()` : Makes g visible, if hidden earlier.
- `left, right, forward` are available as member functions.
- `g.rotate(A)` : Rotate g clockwise. `A` is in radians.
- `g.getX(), g.getY(), g.getOrientation(), g.getScale()` : Returns the specified information about g.
- g.imprint() : An image of g is drawn at the current position. The image will persist even if g moves.
- `g.reset(arg-list)` : Will cause g to be re-initialized using the given `arg-list`, which must have the same form as at creation.

Note: Rotation and scaling cannot be performed on text.

A fun program

IIT Bombay

We will show what happens when a rectangle is tossed up and given a spin.

We will also change its colour as it moves.

Summary

IIT Bombay

We discussed coordinate based graphics as supported in Simplecpp.

Summary

IIT Bombay

We discussed coordinate based graphics as supported in Simplecpp.

- Graphics objects also have an in memory part, which is just a regular object.

Summary

IIT Bombay

We discussed coordinate based graphics as supported in Simplecpp.

- Graphics objects also have an in memory part, which is just a regular object.
- By calling member functions of the in memory part, you can manipulate the appearance on the screen.

Summary

IIT Bombay

We discussed coordinate based graphics as supported in Simplecpp.

- Graphics objects also have an in memory part, which is just a regular object.
- By calling member functions of the in memory part, you can manipulate the appearance on the screen.
- Animation is possible. Rotation, translation, scaling, colour change can be used for interesting effects. The possibilities for doing projects are endless..