CSE115 / CSE503
Introduction to Computer Science I

Dr. Carl Alphonce
343 Davis Hall
alphonce@buffalo.edu

Office hours:
Thursday 12:00 PM – 2:00 PM
Friday 8:30 AM – 10:30 AM
OR request appointment via e-mail
Turn off and put away electronics:

- cell phones
- pagers
- laptops
- tablets
- etc.
Where we’ve been
- primitives
- exam Q&A

Today
- Graphics intro

Tuesday evening
- exam

Wednesday
- make-up exams (NO REGULAR CLASS)

Friday
- exam return
- Interfaces
Exam #1
Tuesday 3/8, 9:15 PM - 10:15 PM

Abayev – Dawkins NSC 201
De Jesus – Geiger NSC 210
Gensel – Jaiswal NSC 215
Jeffrey – Lagerwall NSC 220
Lam – Smalley NSC 225
Soares – Tulenko NSC 218
Valentin – Williams, J. NSC 222
Williams, K. – Zuccala NSC 228
Where was I?
Association for Computing Machinery

SIGACCESS Membership – Accessible Computing
SIGACT Membership – Algorithms and Computation Theory
SIGAda Membership – Ada Programming
SIGAI Membership – Artificial Intelligence
SIGAPP Membership – Applied Computing
SIGARCH Membership – Computer Architecture
SIGBED Membership – Embedded Systems
SIGBio Membership – Bioinformatics, Comp Bio & Biomed Informatics
SIGCAS Membership – Computers and Society
SIGCHI Membership – Computer-Human Interaction
SIGCOMM Membership – Data Communication
SIGCSE Membership – Computer Science Education
SIGDA Membership – Design Automation
SIGDOC Membership – Design of Communication
SIGGEO Membership – Electronic Commerce
SIGEVO Membership – Genetic and Evolutionary Computation
SIGGRAPH Membership – Computer Graphics and Interactive Techniques
SIGHPC Membership – High Performance Computing
SIGIR Membership – Information Retrieval
SIGITE Membership – Information Technology Education
SIGKDD Membership – Knowledge Discovery in Data
SIGLOG Membership – Logic and Computation
SIGMETRICS Membership – Measurement and Evaluation
SIGMICRO Membership – Microarchitecture
SIGMIS Membership – Management Information Systems
SIGMM Membership – Multimedia
SIGMOBILE Membership – Mobility of Systems, Users, Data and Computing
SIGMOD Membership – Management of Data
SIGOPS Membership – Operating Systems
SIGPLAN Print Membership – Programming Languages
SIGSAC Membership – Security, Audit and Control
SIGSAM Membership – Symbolic and Algebraic Manipulation
SIGSIM Membership – Simulation and Modeling
SIGSOFT Membership – Software Engineering
SIGSPATIAL Membership – Special Interest Group on Spatial Information
SIGUCCS Membership – University and College Computing Services
SIGWEB Membership – Hypertext, Hypermedia and Web
Association for Computing Machinery

SIGCSE – Computer Science Education

SIGCSE (1200-1300) (Technical Symposium)

ITiCSE (~300) (Innovation and Technology in CS Education)

ICER (~80) (International Computing Education Research)

DC (~20) (Doctoral Consortium)
Graphical User Interface (GUI)
In these slides we will explain the basics of how to create graphical programs.

Some advanced issues will be glossed over (e.g. thread safety, graphics library design).
There are two basic types of graphical elements:

Containers
- able to hold graphical objects, such as containers and components

Components
- must be put into containers
- able to generate events when manipulated

Containers:
- JFrame
- JButton
Top-level containers
some containers are called “top-level” because they do not need to be placed inside any other containers

JFrame is a top-level container, meaning it can exist independently; a JFrame draws a window, complete with a title bar, scroll-bar, resize controls, etc.

Other containers (not top-level)
most containers must be placed inside some other container
javax.swing.JPanel is an example
Top-level containers have multiple panes

javax.swing.JFrame

image credit: http://docs.oracle.com/javase/tutorial/uiswing/components/toplevel.html
We will add components to the content pane.

With javax.swing.JFrame, two ways:
call getContentPane() on frame to get frame’s content pane, then
call add(...) on content pane to add a component
call add(...) directly on the JFrame object

Second approach is just a convenience method, does the same thing the first approach.
Creating just a frame
new javax.swing.JFrame()

Creating a frame with a title
new javax.swing.JFrame(“My title”)  

Making the frame visible
call setVisible(true) on the frame

Making application close when window is closed:
call setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE) on the frame