COSC 121: Computer Programming II

Dr. Bowen Hui

University of British Columbia Okanagan

Course overview

- Course outline

Course overview

- Course outline
- Today:
  - Course logistics, policies, expectations
  - Concepts and learning outcomes
  - Quick review

What to know before this class

- How to write a class in Java
  - From English description to requirements/design to Java class
  - Class responsibility
  - Attributes, Methods
  - Visibility modifiers
- Basic constructs in Java
- Checking your own work

What to know before this class

- How to write a class in Java
- Basic constructs in Java
  - Variables, data types, assignments, scoping
  - Arithmetic expressions
  - Control flow, method calls
  - Using classes, including Scanner, Random, Math
  - Conditionals, loops, arrays
  - Overloading methods
  - Pass by value vs. pass by reference
- Checking your own work

What to know before this class

- How to write a class in Java
- Basic constructs in Java
- Checking your own work
  - Documenting your code
  - “Tracing” your code
  - Using println statements to check if your code is doing what you think it’s doing

What is this course about?

- How to write “more complicated” programs
  - Good coding practice
  - Class relationships
    - Eliminate redundant code
    - Standardizing how classes are developed
  - Using files
  - Abstract data types/data structures
    - Static vs. dynamic lists
    - Queues and Stacks
  - Algorithms: Sorting and Searching

Caution

- Concepts in this course are much more abstract

- Be sure you are comfortable with the concept, practice lots with actual examples, then move on

- Helps to do pencil-paper examples first
  i.e., don't just dive into the code!

Evaluation Criteria

- 20% Weekly labs (10 total)
- 20% Assignments (3 total)
- 20% Midterm
- 40% Final exam (cumulative)

A lot of work!

- Keep on top of the readings and lectures
- Do practice questions in the text
- Get help AS SOON AS you sense you're falling behind
- Participate actively and interactively
- Practice programming regularly

Late Policy

- Labs and assignments
  - Due at start of lab/class
    - E.g., 10:59AM not 11:00AM
    - Very stringent
  - No lates accepted without a medical note

Missed Exams

- Missed midterms:
  - Receive a mark of 0% without a valid medical note
  - With a valid medical note: missed exam portion will be combined with the final exam (all exams taken will still be worth 60% of the course grade)
- Missed final:
  - Receive a mark of 0% without a valid medical note
  - With a valid medical note accepted by the Dean's Office, a make-up exam will be scheduled

Passing Criteria

- Students must achieve a passing grade in the final exam in order to pass the course
- Students must achieve a passing grade in the lab component in order to pass the course
- Otherwise:
  - Student will receive a max of 45% as the final grade

Expectations in lectures

- Format:
  - Lectures (interaction!!!)
  - Slides & board notes
  - Group activities with TA support
  - Regular “showcase” of cool applications
- You need to:

Expectations in lectures

- Format:
  - Lectures (interaction!!!)
  - Slides & board notes
  - Group activities with TA support
  - Weekly “showcase” of cool applications
- You need to:
  - Take notes (this is not a 39-hour movie)
  - Participate in group activities
  - Listen actively
  - Ask questions
  - Provide answers

Expectations in labs

- Pre-lab exercises
  - Handwritten exercises
- In-lab activities
  - Programming exercises
  - Modify games in each lab
- You need to:

Expectations in labs

- Pre-lab exercises
  - Handwritten exercises
- In-lab activities
  - Programming exercises
  - Modify games in each lab
- You need to:
  - Attend every lab on time
  - Submit pre-lab exercises at beginning of each lab
  - Submit your own work, option to collaborate with others
  - Complete lab activities (one week to submit)

Expectations on assignments

- Mix of written and programming questions
- Submit your own work
- Indicate all collaborations
- You need to:

Expectations on assignments

- Mix of written and programming questions
- Submit your own work
- Indicate all collaborations
- You need to:
  - Work on assignment questions honestly
  - Follow instructions on assignments to get points
  - Submit on time (very stringent)

Expectations on exams

- All individual, handwritten (even programming)
- One 8”x11” cheatsheet allowed
  - Helps organize notes into one spot
  - Process of writing notes reinforces concepts
  - Organization makes relationships more clear
  - Provide (helpful) starting point of reference for your answers
- You need to:

Expectations on exams

- All individual, handwritten (even programming)
- One 8”x11” cheatsheet allowed
  - Helps organize notes into one spot
  - Process of writing notes reinforces concepts
  - Organization makes relationships more clear
  - Provide (helpful) starting point of reference for your answers
- You need to:
  - Study
  - Understand examples from classes, labs, assignments
  - Sleep well the night before the exam

Resources

- Me
- TAs
- Course website *** know this well
- Lab manual
- Course discussion forum on Connect
- Textbook
- Math and Science Tutoring Centre in UNC 201
- Others:
  - You will find many others… we don't necessarily endorse them, but we can't prevent you from using them
  - Use with caution!
  - Don't plagiarize!

Readings

- Today:
  - Review

- Next class:
  - Ch. 9 on inheritance

Lab 1

- Labs start next week
  - Complete pre-lab before arriving to lab
  - Some changes to lab structure – see Guidelines
  - Due: following lab

Templates for Starter Code

- Class template
  - All other classes you write
  - Test class
  - New ones introduced in this course
- Method template
  - Special method: Constructor
  - Special method: toString()
  - Accessors and mutators

Templates for Starter Code

- Class template

  visibility className
  {
  // attributes
  // methods
  }

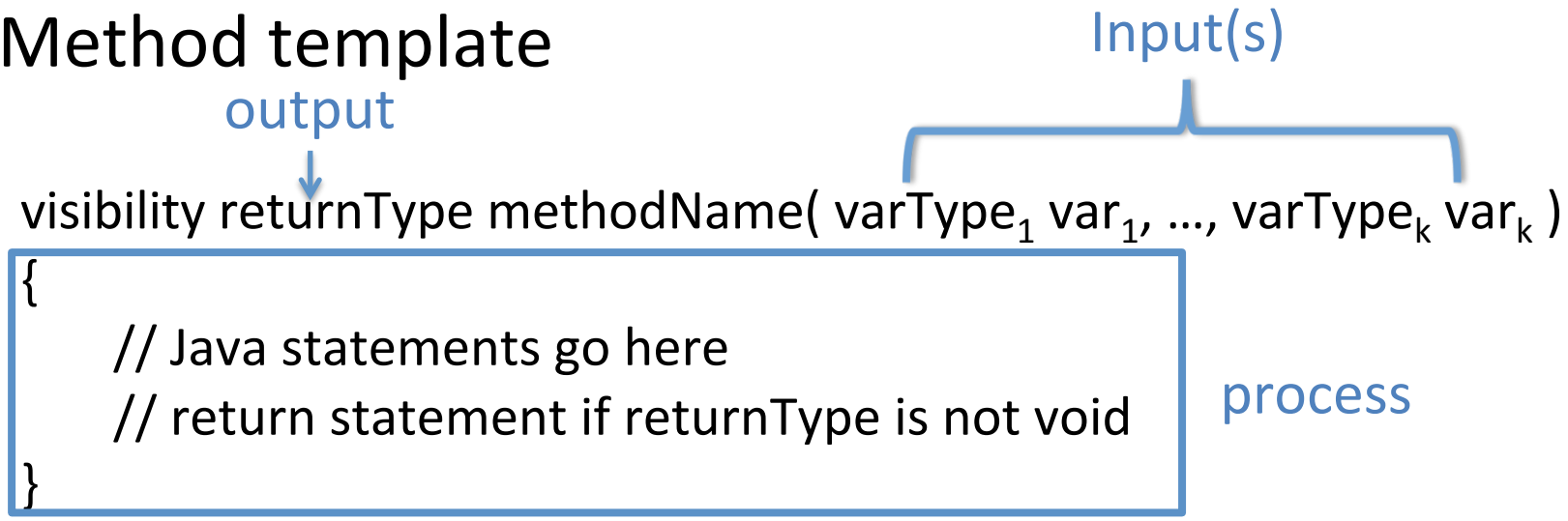
blueprint to define commonalities of its objects

traits about objects
operations objects can do

- Special class:
  - Test class (your text calls this a driver class)
  - This course: others will be introduced

Templates for Starter Code

- Method template



- Special methods:
  - Constructor
  - toString()
  - Accessors and mutators

Quick Review (cont.)

In the following statement, what is the data type specified?

```
int identifier = -931;
```

(a) `int`

(b) `identifier`

(c) `931`

(d) `-931`

(e) None of the above

Quick Review (cont.)

In the following statement, what is the name of the variable?

```
String playerName = "Terry Jones";
```

(a) `String`
(b) `playerName`
(c) `Terry Jones`
(d) `"Terry Jones"`
(e) None of the above

Quick Review (cont.)

In the following code, what is the name of the method?

```
public static void main( String[] args )
{
  // ... missing code
}
```

(a) `args`

(b) `void`

(c) `static`

(d) `main`

(e) `public`

Quick Review (cont.)

In the following statements, what type of object is `postcard`?

```
Post postcard;
postcard = new Post( 52 );
```

(a) `postcard`

(b) `int`

(c) `String`

(d) `Post`

(e) `Player`

Quick Review (cont.)

In the following statement, what is the name of the method?

```
postcard.setAuthor( "Daisy" );
```

(a) `postcard`
(b) `setAuthor`
(c) `"Daisy"`
(d) `Daisy`
(e) None of the above

Quick Review (cont.)

In the following code, how many class attributes does the `TestPost` class have?

```
public class TestPost
{
  public static void main( String[] args )
  {
    String str;
    Post p0;
    Comment c1;

    p0 = new Post( 52 );
    p0.setAuthor( "daisy" );
    str = "A Dog's Life";
    p0.setTitle( str );
    str =  "Being a dog is tough. I sleep all day";
    str += " and work for cookies when Daddy's home.";
    p0.setText( str );
    System.out.println( p0.toString() );
  }
}
```

(a) 0

(b) 1

(c) 2

(d) 3

(e) 4

Quick Review (cont.)

Given the statements below, which of the following boolean expressions evaluates to true?

```
int total = 10;
int MAX   = 7;
boolean found = false;
```

(a) `( total > MAX + 5 && !found )`

(b) `( total > MAX + 5 && found )`

(c) `( total < MAX + 5 && !found )`

(d) `( total < MAX + 5 && found )`

(e) None of the above

Quick Review (cont.)

Given the statements below, which of the following boolean expressions evaluates to false?

```
int x = 5;
int y = 10;
int z = 15;
boolean done = true;
```

(a) `( done || !done )`

(b) `(( x < y )||( z < y ))`

(c) `!( x == y )`

(d) `(( x + 5 ) >= z )`

(e) None of the above

Quick Review (cont.)

How many names will be printed in the following code?

```
String[] names = {"eva", "ann", "cam" };
for( int i=0; i<=names.length; i++ )
  System.out.println( names[i] );
```

(a) 3, but the program will crash with an ArrayIndexOutOfBounds Exception

(b) 3, with no errors

(c) 4, with no errors

(d) Keep printing until you interrupt the program; this is an infinite loop

(e) Cannot be determined

Quick Review (cont.)

In the following statement, how many method calls are there?

```
System.out.println( casper.toString() );
```

(a) 0

(b) 1

(c) 2

(d) 3

(e) 4

Quick Review (cont.)

In the following code, what is the value of y?

```
int i = 5;
for( i=0; i<10; i++ )
  if( i % 2 == 1 )
    break;
int y = i;
```

(a) 0

(b) 1

(c) 2

(d) 3

(e) None of the above

Quick Review (cont.)

What output is printed by the following code?

```
int num1 = 2;
int num2 = 10;
if( num1 < num2 )
  System.out.print( "1 " );
if(( num1 + 5 ) < num2 )
  System.out.print( "2 " );
else
  System.out.print( "3 " );
System.out.print( "4 " );
```

(a) 2 4

(b) 1 2 4

(c) 2 3

(d) 1 2 3 4

(e) 1 2

Quick Review (cont.)

What is the output printed by the following code?

```
int[][] arr2d;
arr2d = new int[3][5];
System.out.println( arr2d.length );
```

(a) 3

(b) 5

(c) 8

(d) 15

(e) 0

Quick Review (cont.)

Consider the following code:

```
int     count = 2;
int     MAX   = 10;
String  xx    = "abba";
String  yy    = "abba";
boolean done  = true;
```

Part b. (1 point) What is the value of the following condition:

```
!!!done
```

Quick Review (cont.)

- Make sure you …
  - Are able to read code given to you (with documentation)
  - Are able to write methods to carry out a specific operation
  - Are able to write classes that have certain responsibilities