

# CS 61B Midterm 1 Review

Dan Wang and Jonathan Kotker

Eta Kappa Nu, Mu Chapter  
University of California, Berkeley

7 October 2011

## What will be printed?

What is printed after the following code is executed?

```
1  public static void main(String[] args) {  
2      String s = "Is this the real life?";  
3      change(s);  
4      System.out.println(s);  
5  }  
6  public static void change(String s) {  
7      s = "Is this just fantasy?";  
8  }
```

1. Is this the real life?
2. Is this just fantasy?
3. s
4. Error

## What will be printed?

What is printed after the following code is executed?

```
1  public static void main(String[] args) {
2      String s = "Is this the real life?";
3      change(s);
4      System.out.println(s);
5  }
6  public static void change(String s) {
7      s = "Is this just fantasy?";
8  }
```

1. Is this the real life?
2. Is this just fantasy?
3. s
4. Error

## What will be printed?

What is printed after the following code is executed?

```
1 public static void main(String[] args) {  
2     int[] arr = {1, 2, 3};  
3     change(arr);  
4     System.out.println(arr[0]);  
5 }  
6 public static void change(int[] i) {  
7     i[0] = 5;  
8     i = null;  
9 }
```

1. 1
2. 5
3. null
4. error

## What will be printed?

What is printed after the following code is executed?

```
1 public static void main(String[] args) {  
2     int[] arr = {1, 2, 3};  
3     change(arr);  
4     System.out.println(arr[0]);  
5 }  
6 public static void change(int[] i) {  
7     i[0] = 5;  
8     i = null;  
9 }
```

1. 1
2. **5**
3. null
4. error

## What will be printed?

What is printed after the following code is executed?

```
1 public static void main(String[] args) {  
2     int herp = 4;  
3     int derp = 6;  
4     herp = derp;  
5     herp = herp + 1;  
6     System.out.println(derp);  
7 }
```

1. 4

2. 6

3. 5

4. 7

## What will be printed?

What is printed after the following code is executed?

```
1 public static void main(String[] args) {  
2     int herp = 4;  
3     int derp = 6;  
4     herp = derp;  
5     herp = herp + 1;  
6     System.out.println(derp);  
7 }
```

1. 4

2. 6

3. 5

4. 7

## What will be printed?

What is printed after the following code is executed?

```
1 public static void main(String[] args) {  
2     String x = "Caught in a landslide,";  
3     String y = "No escape from reality";  
4     String z = x;  
5     x = y;  
6     System.out.println(z);  
7 }
```

1. Caught in a landslide,
2. No escape from reality
3. null
4. Error



## What will be printed?

What is printed after the following code is executed?

```
1 public static void main(String[] args) {  
2     String x = "Caught in a landslide,";  
3     String y = "No escape from reality";  
4     String z = x;  
5     x = y;  
6     System.out.println(z);  
7 }
```

1. Caught in a landslide,
2. **No escape from reality**
3. null
4. Error

## What will be printed?

What is printed after the following code is executed?

```
1  Panda p = new Panda();  
2  Animal a = p;  
3  boolean wat = (a == p);  
4  System.out.println(wat);
```

1. true
2. false
3. wat
4. Error

## What will be printed?

What is printed after the following code is executed?

```
1  Panda p = new Panda();  
2  Animal a = p;  
3  boolean wat = (a == p);  
4  System.out.println(wat);
```

1. **true**
2. false
3. wat
4. Error

## What will happen?

What will happen when the following code is run? Assume that Subclass is a subclass of Class and do\_something is a non-static method in both classes.

```
1  Class c = new Subclass();  
2  c.do_something();
```

1. Class's method is called
2. Subclass's method is called
3. Compile-time error
4. Run-time error

## What will happen?

What will happen when the following code is run? Assume that Subclass is a subclass of Class and do\_something is a non-static method in both classes.

```
1  Class c = new Subclass();  
2  c.do_something();
```

1. Class's method is called
2. Subclass's **method is called**
3. Compile-time error
4. Run-time error

## What will happen?

What will happen when the following code is run? Assume that Subclass is a subclass of Class and do\_something is a non-static method in both classes.

```
1 Subclass c = new Class();  
2  
3 c.do_something();
```

1. Class's method is called
2. Subclass's method is called
3. Compile-time error
4. Run-time error

## What will happen?

What will happen when the following code is run? Assume that Subclass is a subclass of Class and do\_something is a non-static method in both classes.

```
1 Subclass c = new Class();  
2  
3 c.do_something();
```

1. Class's method is called
2. Subclass's method is called
3. **Compile-time error**
4. Run-time error

## What will happen?

What will happen when the following code is run? Assume that Subclass is a subclass of Class and some\_value is a field in both classes.

```
1 Class c = new Subclass();  
2 System.out.println(c.some_value);
```

1. Class's field is printed
2. Subclass's field is printed
3. Compile-time error
4. Run-time error



## What will happen?

What will happen when the following code is run? Assume that Subclass is a subclass of Class and some\_value is a field in both classes.

```
1 Class c = new Subclass();  
2 System.out.println(c.some_value);
```

1. Class's field is printed
2. Subclass's field is printed
3. Compile-time error
4. Run-time error

## What will happen?

What will happen when the following code is run? Assume that `Subclass` is a subclass of `Class` and `static_value` is a **static** field in both classes.

```
1 Class c = new Subclass();  
2 System.out.println(c.static_value);
```

1. Class's field is printed
2. Subclass's field is printed
3. Compile-time error
4. Run-time error

## What will happen?

What will happen when the following code is run? Assume that `Subclass` is a subclass of `Class` and `static_value` is a **static** field in both classes.

```
1 Class c = new Subclass();  
2 System.out.println(c.static_value);
```

1. Class's field is printed
2. Subclass's field is printed
3. Compile-time error
4. Run-time error

## What will happen?

What will happen when the following code is run? Assume that Subclass is a subclass of Class and `static_method()` is a **static** method in both classes.

```
1 Class c = new Subclass();  
2 c.static_method();
```

1. Class's method is called
2. Subclass's method is called
3. Compile-time error
4. Run-time error

## What will happen?

What will happen when the following code is run? Assume that Subclass is a subclass of Class and `static_method()` is a **static** method in both classes.

```
1 Class c = new Subclass();  
2 c.static_method();
```

1. Class's method is called
2. Subclass's method is called
3. Compile-time error
4. Run-time error

## General Rule

In general, if we define a variable `var` as such:

```
1 // S and D are predefined classes
2 S var = new D();
3 S.X;
```

Then `S` is the **static type** of `var` and `D` is the **dynamic type** of `var`. If we attempt to access a field or method of `var`, which one is called?

- If `X` is a **field**, the field from the **static type** of `var` will be used.
- If `X` is a **method**, then it depends on whether or not it is static:
  - If `X` is a **static method**, then the method from the **static type** of `var` will be used
  - If `X` is a **non-static method**, then Java will use dynamic method lookup to determine which class's method to call, starting from the lowest class in the hierarchy.

## Fields

If we have an object of type `Subclass` that extends `Class`, how can we access...

- A field from `Subclass`?
- A field from `Class`?

## Fields

If we have an object of type Subclass that extends Class, how can we access...

- A field from Subclass?

```
1 Subclass s = new Subclass();  
2 s.X;
```

- A field from Class?



## Fields

If we have an object of type Subclass that extends Class, how can we access...

- A field from Subclass?

```
1 Subclass s = new Subclass();  
2 s.X;
```

- A field from Class?

```
1 Class s = new Subclass();  
2 s.X;
```

## Fields

If we have an object of type Subclass that extends Class, how can we access...

- A field from Subclass?

```
1 Subclass s = new Subclass();  
2 s.X;
```

- A field from Class?

```
1 Class s = new Subclass();  
2 s.X;
```

Alternatively, we can cast our variable:

```
1 Subclass s = new Subclass();  
2 ((Class) s).X;
```

## Static Methods

If we have an object of type `Subclass` that extends `Class`, how can we access...

- A static method from `Subclass`?
- A static method from `Class`?

## Static Methods

If we have an object of type Subclass that extends Class, how can we access...

- A static method from Subclass?

```
1 Subclass s = new Subclass();  
2 s.X();
```

- A static method from Class?

## Static Methods

If we have an object of type Subclass that extends Class, how can we access...

- A static method from Subclass?

```
1 Subclass s = new Subclass();  
2 s.X();
```

- A static method from Class?

```
1 Class s = new Subclass();  
2 s.X();
```

## Static Methods

If we have an object of type Subclass that extends Class, how can we access...

- A static method from Subclass?

```
1 Subclass s = new Subclass();  
2 s.X();
```

- A static method from Class?

```
1 Class s = new Subclass();  
2 s.X();
```

Alternatively, we can cast our variable:

```
1 Subclass s = new Subclass();  
2 ((Class)s).X();
```

## Non-static Methods

If we have an object of type Subclass that extends Class, how can we access...

- A non-static method from Subclass, assuming that the method is defined in both Class and Subclass?
- A non-static method from Class, assuming that the method is defined in both Class and Subclass?

## Non-static Methods

If we have an object of type Subclass that extends Class, how can we access...

- A non-static method from Subclass, assuming that the method is defined in both Class and Subclass?

```
1 Subclass s = new Subclass();  
2 s.Y();
```

- A non-static method from Class, assuming that the method is defined in both Class and Subclass?



## Non-static Methods

If we have an object of type Subclass that extends Class, how can we access...

- A non-static method from Subclass, assuming that the method is defined in both Class and Subclass?

```
1 Subclass s = new Subclass();  
2 s.Y();
```

- A non-static method from Class, assuming that the method is defined in both Class and Subclass?

This is impossible! This is a feature of Java, not a bug. When you override a non-static method in your parent class, you are specifying a *more specific* action for your subclass to take. If you require the original behaviour of the parent class's method, it is much better design to create another method.

## Remove Duplicates

Write a function `removeDuplicates()` that takes in an `IntList` and *destructively* removes all duplicate items without using any other data structures (no arrays or other `IntLists`!). The `IntList` headers are shown below:

```
1  class IntList {  
2      public int head;  
3      public IntList tail;  
4  
5      public IntList(int head, IntList tail) {  
6          ...  
7      }  
8  }
```

You may use helper methods if you wish. Your method should have the following header:

```
1  public static void removeDuplicates(IntList list) {  
2      ...  
3  }
```

## Remove Duplicates

Solution:

```
1  public static void removeDuplicates(IntList list) {
2      IntList current = list;
3      while(current != null) {
4          int value = current.head;
5          IntList l = current;
6          while(l.tail != null) {
7              if(l.tail.head == value) {
8                  l.tail = l.tail.tail;
9              } else {
10                 l = l.tail;
11             }
12         }
13         current = current.tail;
14     }
15 }
```

## Remove Duplicates

Solution:

```
1  public static void removeDuplicates(IntList list) {  
2      IntList current = list;  
3      while(current != null) {  
4          int value = current.head;  
5          IntList l = current;  
6          while(l.tail != null) {  
7              if(l.tail.head == value) {  
8                  l.tail = l.tail.tail;  
9              } else {  
10                 l = l.tail;  
11             }  
12         }  
13         current = current.tail;  
14     }  
15 }
```

What is the runtime of this algorithm in terms of  $n$ , the length of the input `IntList`?

## Remove Duplicates

Solution:

```
1  public static void removeDuplicates(IntList list) {
2      IntList current = list;
3      while(current != null) {
4          int value = current.head;
5          IntList l = current;
6          while(l.tail != null) {
7              if(l.tail.head == value) {
8                  l.tail = l.tail.tail;
9              } else {
10                 l = l.tail;
11             }
12         }
13         current = current.tail;
14     }
15 }
```

What is the runtime of this algorithm in terms of  $n$ , the length of the input `IntList`?

$$O(n^2)$$

## List Reversal

Write a function `reverse()` that takes in an `IntList` and *destructively* removes all duplicate items without using any other data structures (no arrays or other `IntLists`!). The `IntList` headers are shown below:

```
1  class IntList {
2      public int head;
3      public IntList tail;
4
5      public IntList(int head, IntList tail) {
6          ...
7      }
8  }
```

You may use helper methods if you wish. Your method should have the following header:

```
1  public static void reverse(IntList list) {
2      ...
3  }
```

## List Reversal

Solution:

```
1  public static void reverse(IntList list) {  
2      IntList reversed = null;  
3      while(list != null) {  
4          IntList temp = list;  
5          list = list.tail;  
6          temp.tail = reversed;  
7          reversed = temp;  
8      }  
9  }
```

## List Reversal

Solution:

```
1  public static void reverse(IntList list) {  
2      IntList reversed = null;  
3      while(list != null) {  
4          IntList temp = list;  
5          list = list.tail;  
6          temp.tail = reversed;  
7          reversed = temp;  
8      }  
9  }
```

What is the runtime of this algorithm in terms of  $n$ , the length of the input `IntList`?



## List Reversal

Solution:

```
1 public static void reverse(IntList list) {  
2     IntList reversed = null;  
3     while(list != null) {  
4         IntList temp = list;  
5         list = list.tail;  
6         temp.tail = reversed;  
7         reversed = temp;  
8     }  
9 }
```

What is the runtime of this algorithm in terms of  $n$ , the length of the input `IntList`?

$$O(n)$$

## True or False

For each of the following statements, determine whether it is true or false.

1.  $\log^3 n \in O(n)$
2.  $2^n \in O(n^2)$
3.  $n! \in O(n^n)$
4.  $\sin n \in O(\log n)$
5.  $1 \in O(\frac{1}{n})$

## True or False

For each of the following statements, determine whether it is true or false.

1.  $\log^3 n \in O(n)$  **True.**
2.  $2^n \in O(n^2)$
3.  $n! \in O(n^n)$
4.  $\sin n \in O(\log n)$
5.  $1 \in O(\frac{1}{n})$

## True or False

For each of the following statements, determine whether it is true or false.

1.  $\log^3 n \in O(n)$  True.
2.  $2^n \in O(n^2)$  False.
3.  $n! \in O(n^n)$
4.  $\sin n \in O(\log n)$
5.  $1 \in O(\frac{1}{n})$

## True or False

For each of the following statements, determine whether it is true or false.

1.  $\log^3 n \in O(n)$  True.
2.  $2^n \in O(n^2)$  False.
3.  $n! \in O(n^n)$  True.
4.  $\sin n \in O(\log n)$
5.  $1 \in O(\frac{1}{n})$

## True or False

For each of the following statements, determine whether it is true or false.

1.  $\log^3 n \in O(n)$  True.
2.  $2^n \in O(n^2)$  False.
3.  $n! \in O(n^n)$  True.
4.  $\sin n \in O(\log n)$  True.
5.  $1 \in O(\frac{1}{n})$

## True or False

For each of the following statements, determine whether it is true or false.

1.  $\log^3 n \in O(n)$  True.
2.  $2^n \in O(n^2)$  False.
3.  $n! \in O(n^n)$  True.
4.  $\sin n \in O(\log n)$  True.
5.  $1 \in O(\frac{1}{n})$  False.

## Exercises

Give the tightest upper bound for the following expressions in big-Oh notation.

1.  $1 + 2 + \cdots + n$
2.  $n^2 + 1000$
3.  $\sum_{i=0}^n \sum_{j=i}^n 1$
4.  $n \sin n$
5.  $n + \log^{9001} n$
6.  $\log x^3$
7.  $\frac{1}{n}$



## Exercises

Give the tightest upper bound for the following expressions in big-Oh notation.

1.  $1 + 2 + \cdots + n \in O(n^2)$

2.  $n^2 + 1000$

3.  $\sum_{i=0}^n \sum_{j=i}^n 1$

4.  $n \sin n$

5.  $n + \log^{9001} n$

6.  $\log x^3$

7.  $\frac{1}{n}$

## Exercises

Give the tightest upper bound for the following expressions in big-Oh notation.

1.  $1 + 2 + \cdots + n \in O(n^2)$

2.  $n^2 + 1000 \in O(n^2)$

3.  $\sum_{i=0}^n \sum_{j=i}^n 1$

4.  $n \sin n$

5.  $n + \log^{9001} n$

6.  $\log x^3$

7.  $\frac{1}{n}$

## Exercises

Give the tightest upper bound for the following expressions in big-Oh notation.

1.  $1 + 2 + \cdots + n \in O(n^2)$
2.  $n^2 + 1000 \in O(n^2)$
3.  $\sum_{i=0}^n \sum_{j=i}^n 1 \in O(n^2)$
4.  $n \sin n$
5.  $n + \log^{9001} n$
6.  $\log x^3$
7.  $\frac{1}{n}$

## Exercises

Give the tightest upper bound for the following expressions in big-Oh notation.

1.  $1 + 2 + \cdots + n \in O(n^2)$

2.  $n^2 + 1000 \in O(n^2)$

3.  $\sum_{i=0}^n \sum_{j=i}^n 1 \in O(n^2)$

4.  $n \sin n \in O(n)$

5.  $n + \log^{9001} n$

6.  $\log x^3$

7.  $\frac{1}{n}$

## Exercises

Give the tightest upper bound for the following expressions in big-Oh notation.

1.  $1 + 2 + \cdots + n \in O(n^2)$

2.  $n^2 + 1000 \in O(n^2)$

3.  $\sum_{i=0}^n \sum_{j=i}^n 1 \in O(n^2)$

4.  $n \sin n \in O(n)$

5.  $n + \log^{9001} n \in O(n)$

6.  $\log x^3$

7.  $\frac{1}{n}$

## Exercises

Give the tightest upper bound for the following expressions in big-Oh notation.

1.  $1 + 2 + \cdots + n \in O(n^2)$
2.  $n^2 + 1000 \in O(n^2)$
3.  $\sum_{i=0}^n \sum_{j=i}^n 1 \in O(n^2)$
4.  $n \sin n \in O(n)$
5.  $n + \log^{9001} n \in O(n)$
6.  $\log x^3 \in O(\log x)$
7.  $\frac{1}{n}$

## Exercises

Give the tightest upper bound for the following expressions in big-Oh notation.

1.  $1 + 2 + \cdots + n \in O(n^2)$
2.  $n^2 + 1000 \in O(n^2)$
3.  $\sum_{i=0}^n \sum_{j=i}^n 1 \in O(n^2)$
4.  $n \sin n \in O(n)$
5.  $n + \log^{9001} n \in O(n)$
6.  $\log x^3 \in O(\log x)$
7.  $\frac{1}{n} \in O(1)$