

Software 1 with Java

Recitation No. 13 (Summary)

January 28, 2006

Ohad Barzilay and Oranit Dror

1

Initialization

```
public class Foo {
    static int bar;

    public static void main (String args []) {
        bar += 1;
        System.out.println("bar = " + bar);
    }
}
```

Does the code compile? If no, why?
Does the code throw a runtime exception?
If yes, why? If no, what is the output?

January 28, 2006

Ohad Barzilay and Oranit Dror

2

Initialization

```
public class Test {
    private int a = getB();
    private int b = 5;

    private int getB() {
        return b;
    }

    public static void main(String args[]) {
        System.out.println((new Test()).a);
    }
}
```

Does the code compile? If no, why?
Does the code throw a runtime exception?
If yes, why? If no, what is the output?

January 28, 2006

Ohad Barzilay and Oranit Dror

3

Initialization

```
public class Test {
    private int b = 5;
    private int a = getB();

    private int getB() {
        return b;
    }

    public static void main(String args[]) {
        System.out.println((new Test()).a);
    }
}
```

Does the code compile? If no, why?
Does the code throw a runtime exception?
If yes, why? If no, what is the output?

January 28, 2006

Ohad Barzilay and Oranit Dror

4

Exceptions

```
int i=1, j=1;
try {
    i++;
    j--;
    if (i/j > 1)
        i++;
} catch(ArithmeticException e) {
    System.out.println(1);
} catch(Exception e) {
    System.out.println(2);
} finally {
    System.out.println(3);
}
```

What is the output?

January 28, 2006

Ohad Barzilay and Oranit Dror

5

Pass by Value

```
public class PassTest1 {

    public static void changeInt(int value) {
        value = 55;
    }

    public static void main(String args[]) {
        int val;

        // Assign the int
        val = 11;
        // Try to change it
        changeInt(val);
        // What is the current value?
        System.out.println(val);
    }
}
```

Does the code compile? If no, why?
Does the code throw a runtime exception?
If yes, why? If no, what is the output?

January 28, 2006

Ohad Barzilay and Oranit Dror

6

Pass by Reference

```
public class PassTest2 {  
  
    public static void changeObjectRef(MyPoint ref) {  
        ref = new MyPoint(1, 1);  
    }  
  
    public static void main(String args[]) {  
        MyPoint point;  
        // Assign the point  
        point = new MyPoint(22, 7);  
        // Try to change it  
        changeObjectRef(point);  
        // What is the current value?  
        System.out.println(point);  
    }  
}
```

```
public class MyPoint {  
    int x;  
    int y;  
  
    public MyPoint(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
  
    @Override  
    public String toString() {  
        return "(" + x + "," + y + ")";  
    }  
}
```

Does the code compile? If no, why?
Does the code throw a runtime exception?
If yes, why? If no, what is the output?

January 28, 2006

Ohad Barzilay and Oranit Dror

7

Pass by Reference

```
public class PassTest3 {  
  
    public static void changeObjectAttr(MyPoint ref) {  
        ref.setX(4);  
    }  
  
    public static void main(String args[]) {  
        MyPoint point;  
        // Assign the point  
        point = new MyPoint(22, 7);  
  
        changeObjectAttr(point);  
  
        // What is the current value?  
        System.out.println(point);  
    }  
}
```

```
public class MyPoint {  
    int x;  
    int y;  
  
    public MyPoint(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
  
    public void setX(int x) {  
        this.x = x;  
    }  
  
    @Override  
    public String toString() {  
        return "(" + x + "," + y + ")";  
    }  
}
```

Does the code compile? If no, why?
Does the code throw a runtime exception?
If yes, why? If no, what is the output?

January 28, 2006

Ohad B.

Pass By-Value/Reference

```
public class Test {  
    private static class Value { int v = 1; }  
  
    public static void main(String[] args) {  
        Test test = new Test();  
        int v = 2;  
        Value value = new Value();  
        value.v = 3;  
        foo(value, v);  
        System.out.println(value.v + " " + v);  
    }  
  
    private static void foo(Value value, int v) {  
        v = 4;  
        value.v = 5;  
        value = new Value();  
        System.out.println(value.v + " " + v);  
    }  
}
```

What is the output?

9

Visibility

```
public class A {  
    private int bar = 0;  
  
    public boolean isEqual(A a) {  
        return (bar == a.bar);  
    }  
  
    public static void main(String[] args) {  
        A a1 = new A();  
        A a2 = new A();  
        System.out.println(a1.isEqual(a2));  
    }  
}
```

Does the code compile? If no, why?
Does the code throw a runtime exception?
If yes, why? If no, what is the output?

January 28, 2006

Ohad Barzilay and Oranit Dror

10

Interfaces

```
public interface Foo {  
    public void bar()  
        throws Exception;  
}  
  
public class FooImpl implements Foo {  
    public void bar() {  
        System.out.println("An exception is not thrown");  
    }  
  
    public static void main(String args[]) {  
        Foo foo = new FooImpl();  
        foo.bar();  
    }  
}
```

Does the code compile? If no, why?
Does the code throw a runtime exception?
If yes, why? If no, what is the output?

11

Interfaces

```
public interface Foo {  
    public void bar()  
        throws Exception;  
}  
  
public class FooImpl implements Foo {  
    public void bar() {  
        System.out.println("No exception is thrown");  
    }  
  
    public static void main(String args[]) {  
        FooImpl foo = new FooImpl();  
        foo.bar();  
    }  
}
```

Does the code compile? If no, why?
Does the code throw a runtime exception?
If yes, why? If no, what is the output?

12

Interfaces and Inheritance

Consider the following class hierarchy:

```
Interface Animal {...}
class Dog implements Animal {...}
class Poodle extends Dog {...}
class Labrador extends Dog {...}
```

Which of the following lines (if any) will not compile?

```
Poodle poodle = new Poodle();
Animal animal = (Animal) poodle;
Dog dog = new Labrador();
animal = dog;
poodle = dog;
```

January 28, 2006

Ohad Barzilay and Oranit Dror

13

Interfaces and Inheritance

```
class A {
    void print() {
        System.out.println("A");
    }
}

class B extends A implements C {
}

interface C {
    void print();
}
```

Is there any error?

January 28, 2006

Ohad Barzilay and Oranit Dror

14

Interfaces and Inheritance

```
class A {
    public void print() {
        System.out.println("A");
    }
}

class B extends A implements C {
}

interface C {
    void print();
}
```

Is there any error?

January 28, 2006

Ohad Barzilay and Oranit Dror

15

Method Overloading & Overriding

```
public class A {
    public float foo(float a, float b) throws IOException {
    }
}

public class B extends A {
    ...
}
```

Which of the following methods can be defined in B:

1. float foo(float a, float b){...}
2. public int foo(int a, int b) throws Exception {...}
3. public float foo(float a, float b) throws Exception {...}
4. public float foo(float p, float q){...}

January 28, 2006

Ohad Barzilay and Oranit Dror

16

Method Overloading

```
public class A {
    public void foo(Object o) {
        System.out.println("Object");
    }

    public void foo(String s) {
        System.out.println("String");
    }

    public static void main(String args[]) {
        A a = new A();
        a.foo(null);
    }
}
```

•Does the code compile? If no, why?
•Does the code throw a runtime exception?
If yes, why? If no, what is the output?

January 28, 2006

Ohad Barzilay and Oranit Dror

17

Method Overloading

```
public class A {
    private static class B {}
    private static class C extends B {}
    public void foo(B b) {
        System.out.println("B");
    }

    public void foo(C c) {
        System.out.println("C");
    }

    public static void main(String args[]) {
        A a = new A();
        a.foo(null);
    }
}
```

•Does the code compile? If no, why?
•Does the code throw a runtime exception?
If yes, why? If no, what is the output?

January 28, 2006

Ohad Barzilay and Oranit Dror

18

Method Overloading

```
public class A {
    public void foo(StringBuffer sb) {
        System.out.println("StringBuffer");
    }

    public void foo(String s) {
        System.out.println("String");
    }

    public static void main(String args[]) {
        A a = new A();
        a.foo(null);
    }
}
```

•Does the code compile? If no, why?
•Does the code throw a runtime exception?
If yes, why? If no, what is the output?

January 28, 2006

Ohad Barzilai and Oranit Dror

19

Method Overriding

```
public class A {
    public void print() {
        System.out.println("A");
    }
}

public class B extends A {
    public void print() {
        System.out.println("B");
    }
}
```

```
public class C {
    public static void main(String args[]) {
        B b = new B();
        A a = b;

        b.print();
        a.print();
    }
}
```

Does the code compile? If no, why?
Does the code throw a runtime exception?
If yes, why? If no, what is the output?

January 28, 2006

Ohad B

Method Overriding & Visibility

```
public class A {
    public void print() {
        System.out.println("A");
    }
}

public class B extends A {
    protected void print() {
        System.out.println("B");
    }
}
```

```
public class C {
    public static void main(String[] args) {
        B b = new B();
        b.print();
    }
}
```

Does the code compile? If no, why?
Does the code throw a runtime exception?
If yes, why? If no, what is the output?

January 28, 2006

Ohad Barzilai and Oranit Dror

21

Method Overriding & Visibility

```
public class A {
    protected void print() {
        System.out.println("A");
    }
}

public class B extends A {
    public void print() {
        System.out.println("B");
    }
}
```

```
public class C {
    public static void main(String[] args) {
        B b = new B();
        b.print();
    }
}
```

What is the output?

January 28, 2006

Ohad Barzilai and Oranit Dror

22

Inheritance

```
public class A {
    private void foo() {
        System.out.println("A.foo()");
    }

    public void bar() {
        System.out.println("A.bar()");
        foo();
    }
}

public class B extends A {
    public void foo() {
        System.out.println("B.foo()");
    }
}
```

```
public class D {
    public static void main(String[] args) {
        A a = new B();
        a.bar();
    }
}
```

Does the code compile? If no, why?
Does the code throw a runtime exception?
If yes, why? If no, what is the output?

January 28, 2006

Ohad

Inheritance

```
public class A {
    public void foo() {...}
}

public class B extends A {
    public void foo() {...}
}
```

How can you invoke the foo method of A within B?

January 28, 2006

Ohad Barzilai and Oranit Dror

24

Inheritance

```
public class A {
    public void foo() {...}
}

public class B extends A {
    public void foo() {...}
}

public class C extends B {
    public void foo() {...}
}
```

How can you invoke the foo method of A within C?

January 28, 2006

Ohad Barzilay and Oranit Dror

25

Inheritance & Constructors

```
public class A {
    String bar = "A.bar";
    A() { foo(); }
    public void foo() {
        System.out.println("A.foo(): bar = " + bar);
    }
}

public class B extends A {
    String bar = "B.bar";
    B() { foo(); }
    public void foo() {
        System.out.println("B.foo(): bar = " + bar);
    }
}

public class D {
    public static void main(String[] args) {
        A a = new B();
        System.out.println("a.bar = " + a.bar);
        a.foo();
    }
}
```

What is the output?

January 28, 2006

Ohad Barzilay and Oranit Dror

26

Inheritance & Constructors

```
public class A {
    protected B b = new B();
    public A() { System.out.println("in A: no args."); }
    public A(String s) { System.out.println("in A: s = " + s); }
}

public class B {
    public B() { System.out.println("in B: no args."); }
}

class C extends A {
    protected B b;
    public C() { System.out.println("in C: no args."); }
    public C(String s) { System.out.println("in C: s = " + s); }
}

public class D {
    public static void main(String args[]) {
        C c = new C();
        A a = new C();
    }
}
```

What is the output?

27

Inheritance & Constructors

```
public class A {
    protected B b = new B();
    public A() { System.out.println("in A: no args."); }
    public A(String s) { System.out.println("in A: s = " + s); }
}

public class B {
    public B() { System.out.println("in B: no args."); }
}

public class C extends A {
    protected B b;
    public C() { System.out.println("in C: no args."); }
    public C(String s) { System.out.println("in C: s = " + s); }
}

public class D {
    public static void main(String args[]) {
        C c = new C("e");
        A a = new C("a");
    }
}
```

What is the output?

28

Inheritance & Constructors

```
public class A {
    String bar = "A.bar";
}

public class B extends A {
    String bar = "B.bar";
    B() { foo(); }
    public void foo() {
        System.out.println("B.foo(): bar = " + bar);
    }
}

public class D {
    public static void main(String[] args) {
        A a = new B();
        System.out.println(a.bar);
        a.foo();
    }
}
```

What is the result?

January 28, 2006

Ohad Barzilay and Oranit Dror

29

Local Class


```
public class Test {
    public int a = 0;
    private int b = 1;

    public void foo(final int c) {
        int d = 2;

        class InnerTest {
            private void bar(int e) {
                // highlighted line
            }
        }
    }
}
```

Which variables (a, b, c, d, e) are accessible at the highlighted line?

30



Good-Luck!!!

January 28, 2008

Ohad Barzilay and Oranit Dror

31