

---

# Announcement

- Assignment 4 posted

---

## Section 1: Multiple choice

**Question 1** Which of the following CPU components is responsible for directing the “fetch-decode-execute” cycle:

- a) The ALU
- b) The program counter
- c) The instruction register
- d) The control unit
- e) The decoder

---

## Section 1: Multiple choice

**Question 1** Which of the following CPU components is responsible for directing the “fetch-decode-execute” cycle:

- a) The ALU
- b) The program counter
- c) The instruction register
- d) The control unit
- e) The decoder

Answer: d)

---

## Section 1: Multiple choice

**Question 2** What kind of information cannot be represented by bit sequences?

- a) Text
- b) Numbers
- c) Program instructions
- d) Images
- e) None of the above: all of these can be represented by bit sequences

---

## Section 1: Multiple choice

**Question 2** What kind of information cannot be represented by bit sequences?

- a) Text
- b) Numbers
- c) Program instructions
- d) Images
- e) None of the above: all of these can be represented by bit sequences

Answer: e)

---

## Section 1: Multiple choice

**Question 3** What will be the value of `z` if we execute the statement `double z = 5 / 10;`?

- a) 0.0
- b) 0.5
- c) 5.0
- d) 0.05
- e) None of the above. A run-time error arises because `z` is a double and `5/10` is an int.

---

## Section 1: Multiple choice

**Question 3** What will be the value of  $z$  if we execute the statement `double z = 5 / 10;`?

- a) 0.0
- b) 0.5
- c) 5.0
- d) 0.05
- e) None of the above. A run-time error arises because  $z$  is a double and  $5/10$  is an int.

Answer: a)

---

## Section 1: Multiple choice

**Question 4** Given the following assignment statement, in which order will the operators be applied?

$$a = (b + c) * d / e - f;$$

- a) \*, /, +, -
- b) \*, +, /, -
- c) +, \*, /, -
- d) +, /, \*, -
- e) +, -, \*, /



---

## Section 1: Multiple choice

**Question 4** Given the following assignment statement, in which order will the operators be applied?

$$a = (b + c) * d / e - f;$$

- a) \*, /, +, -
- b) \*, +, /, -
- c) +, \*, /, -
- d) +, /, \*, -
- e) +, -, \*, /

Answer: c)

---

## Section 1: Multiple choice

**Question 5** Consider the following program fragment:

```
int x,y,z;  
x = 3;  
y = x;  
z = y + 2;  
x = x + 5;  
y = y - 1;
```

After executing this fragment, what are the values of x, y and z respectively?

- a) 7, 2 and 4
- b) 7, 7 and 9
- c) 8, 8 and 10
- d) 8, 2 and 5
- e) This fragment has a contradiction, so it generates an error.

---

## Section 1: Multiple choice

**Question 5** Consider the following program fragment:

```
int x,y,z;  
x = 3;  
y = x;  
z = y + 2;  
x = x + 5;  
y = y - 1;
```

After executing this fragment, what are the values of x, y and z respectively?

- a) 7, 2 and 4
- b) 7, 7 and 9
- c) 8, 8 and 10
- d) 8, 2 and 5
- e) This fragment has a contradiction, so it generates an error.

Answer: d)

---

## Section 1: Multiple choice

**Question 6** Which of the following is a syntactically incorrect statement?

- a) `if (true == false) { b = true; b = false; }`
- b) `snowy = !snowy;`
- c) `while (person.isAwake()) { brain.work(); }`
- d) `firstName.charAt(0) = 'E';`
- e) `easy_exam = did_assignments && studied`

`|| 100 != 100 && 100 == 100;`

---

## Section 1: Multiple choice

**Question 6** Which of the following is a syntactically incorrect statement?

- a) `if (true == false) { b = true; b = false; }`
- b) `snowy = !snowy;`
- c) `while (person.isAwake()) { brain.work(); }`
- d) `firstName.charAt(0) = 'E';`
- e) `easy_exam = did_assignments && studied`

`|| 100 != 100 && 100 == 100;`

Answer: d)

---

## Section 1: Multiple choice

**Question 7** What will be printed if the following program fragment is executed?

```
boolean a = true, b = false;
if (a == false) {
    a = true;
    System.out.print("Up");
}
else
    if (b == false) {
        b = false;
        System.out.println("Down");
    }
    else
        System.out.println("Middle");
a = false;
b = true;
```

---

## Section 1: Multiple choice

- a) Up
- b) Down
- c) Middle
- d) Nothing
- e) The program will not compile

Answer: b)

---

## Section 1: Multiple choice

**Question 8** What will be printed if the following program fragment is executed?

```
int p, q, r;
p = 4;
r = 10;
while (p < r) {
    q = p;
    p = p + q / 2;
}
System.out.println(p);
```

- a) 13.5   b) 14   c) 13   d) 9   e) Nothing, it doesn't terminate



---

## Section 1: Multiple choice

**Question 8** What will be printed if the following program fragment is executed?

```
int p, q, r;  
p = 4;  
r = 10;  
while (p < r) {  
    q = p;  
    p = p + q / 2;  
}  
System.out.println(p);
```

- a) 13.5   b) 14   c) 13   d) 9   e) Nothing, it doesn't terminate

Answer: c)

| iteration | p  | q | r  | p<r   |
|-----------|----|---|----|-------|
| 0         | 4  |   | 10 | true  |
| 1         | 6  | 4 | 10 | true  |
| 2         | 9  | 6 | 10 | true  |
| 3         | 13 | 9 | 10 | false |

---

## Section 1: Multiple choice

**Question 9** What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}

public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice

- a) h, b
- b) e, b
- c) h, o
- d) e, o
- e) o, e

---

## Section 1: Multiple choice

- a) h, b
- b) e, b
- c) h, o
- d) e, o
- e) o, e

Answer: b)

---

## Section 1: Multiple choice

**Question 9** What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}

public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice

p frame

|   |                      |
|---|----------------------|
| a | <input type="text"/> |
| b | <input type="text"/> |

---

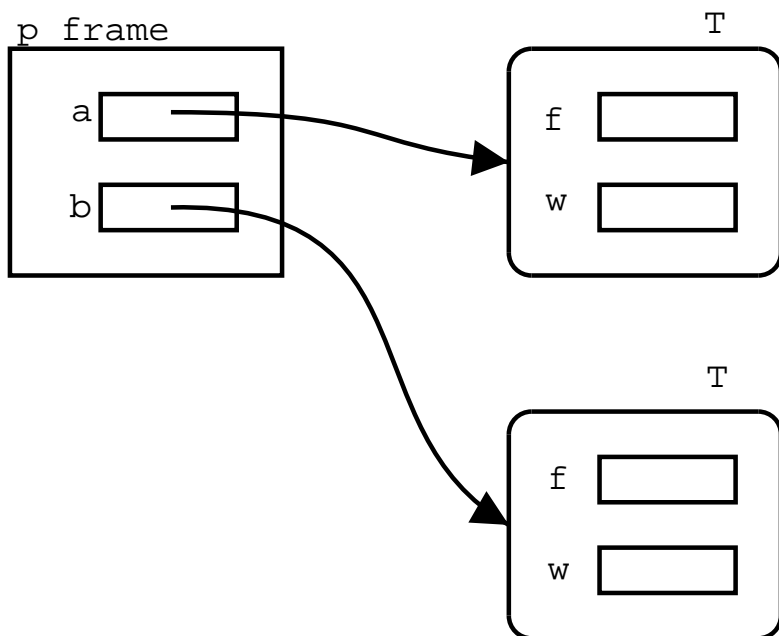
## Section 1: Multiple choice

Question 9 What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}
public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice





---

## Section 1: Multiple choice

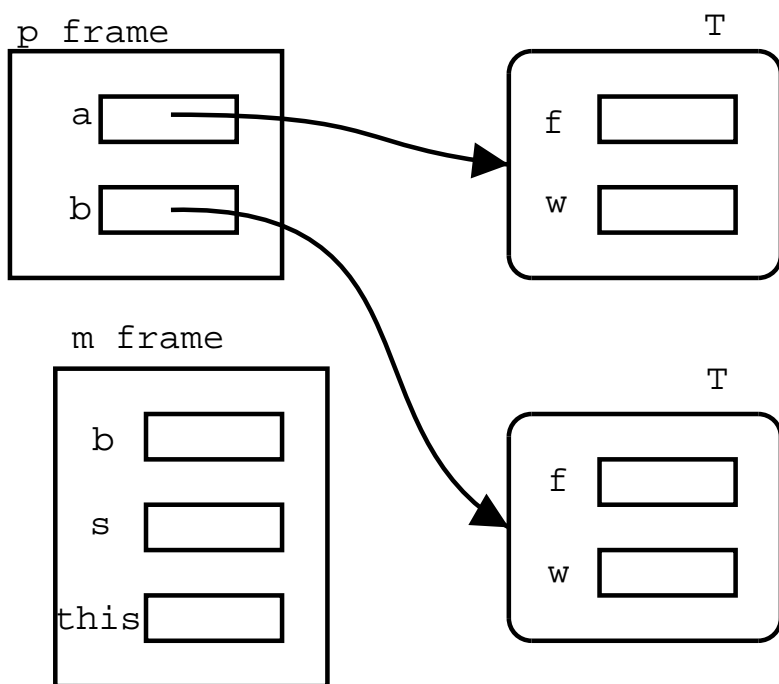
Question 9 What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}

public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice



---

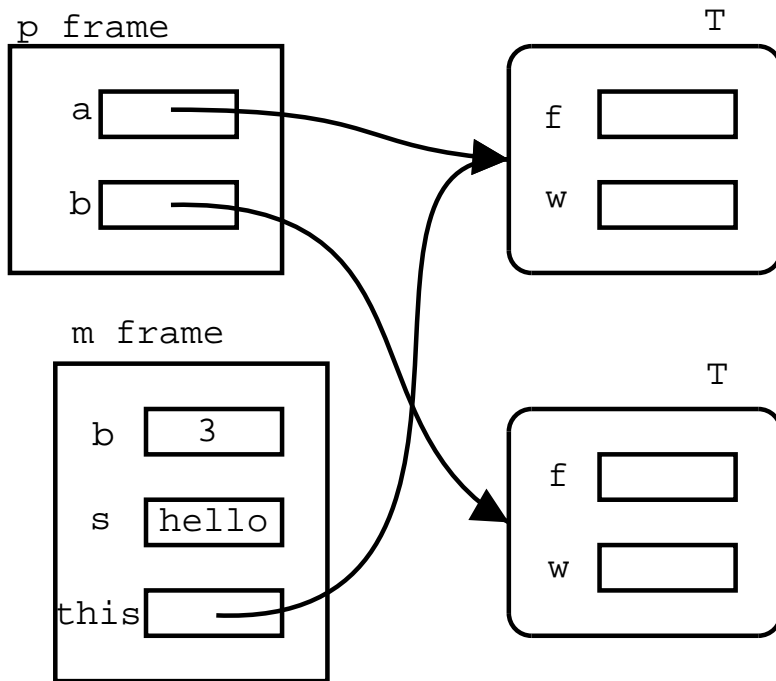
## Section 1: Multiple choice

Question 9 What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}
public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice



---

## Section 1: Multiple choice

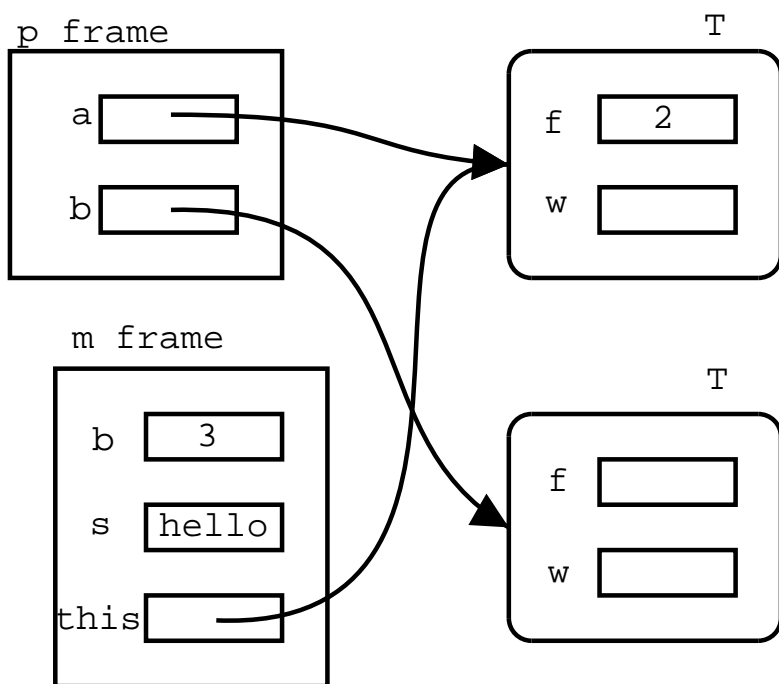
Question 9 What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}

public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice



---

## Section 1: Multiple choice

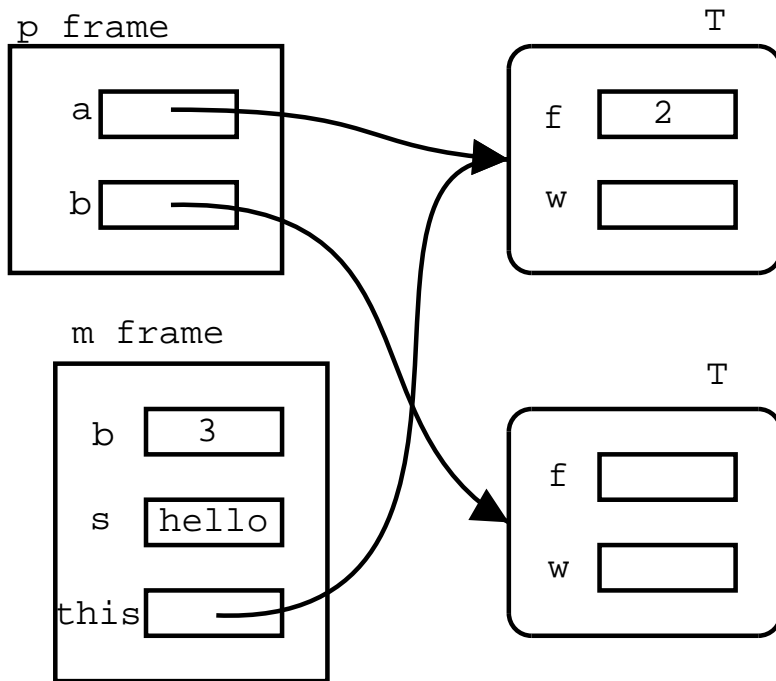
Question 9 What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}

public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice





---

## Section 1: Multiple choice

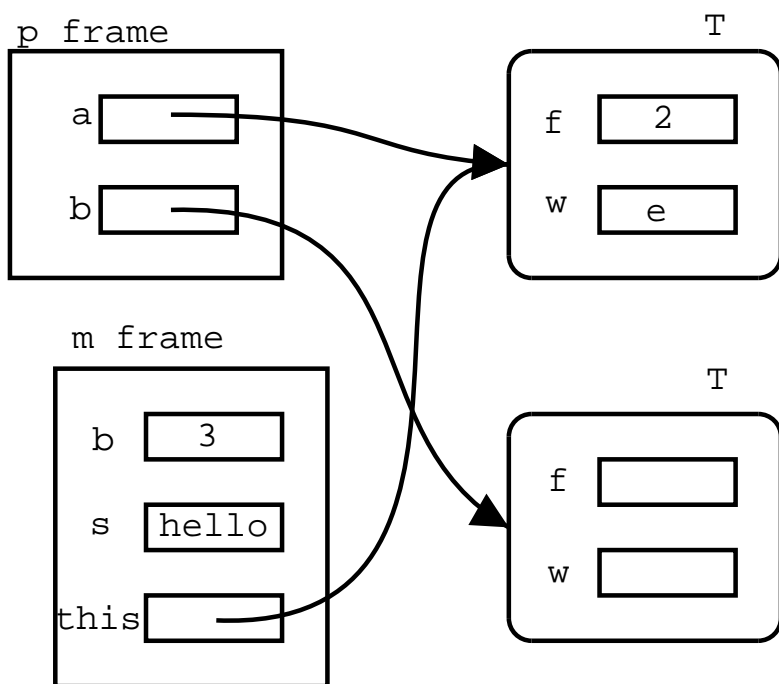
Question 9 What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}

public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice



---

## Section 1: Multiple choice

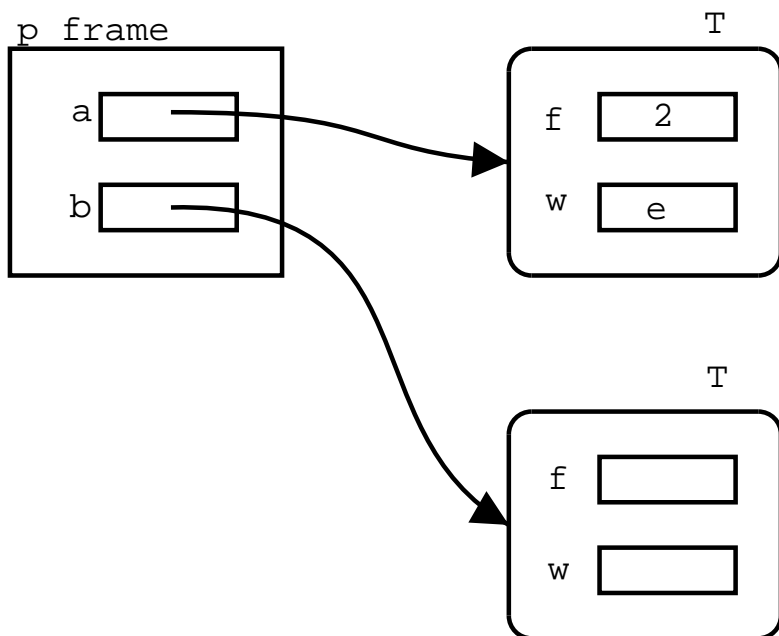
Question 9 What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}

public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice



---

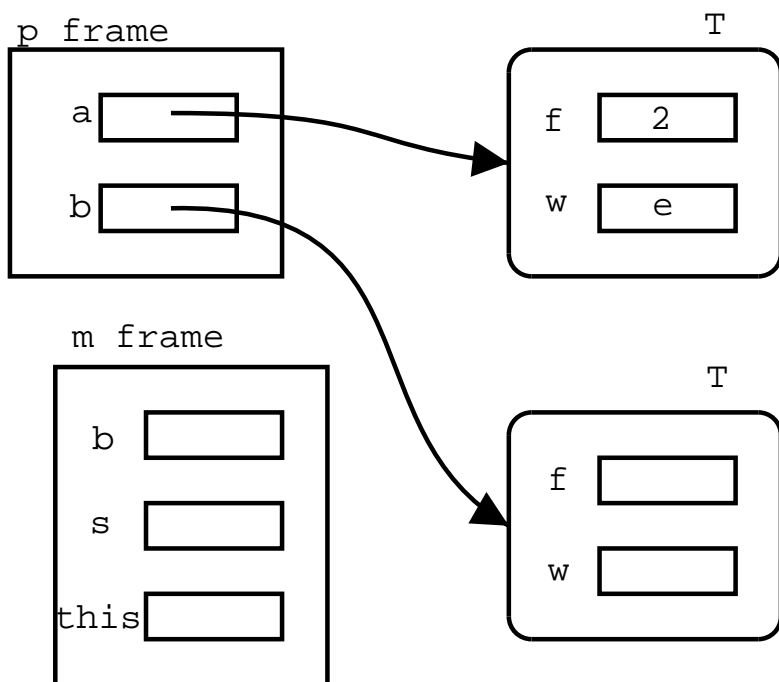
## Section 1: Multiple choice

Question 9 What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}
public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice



---

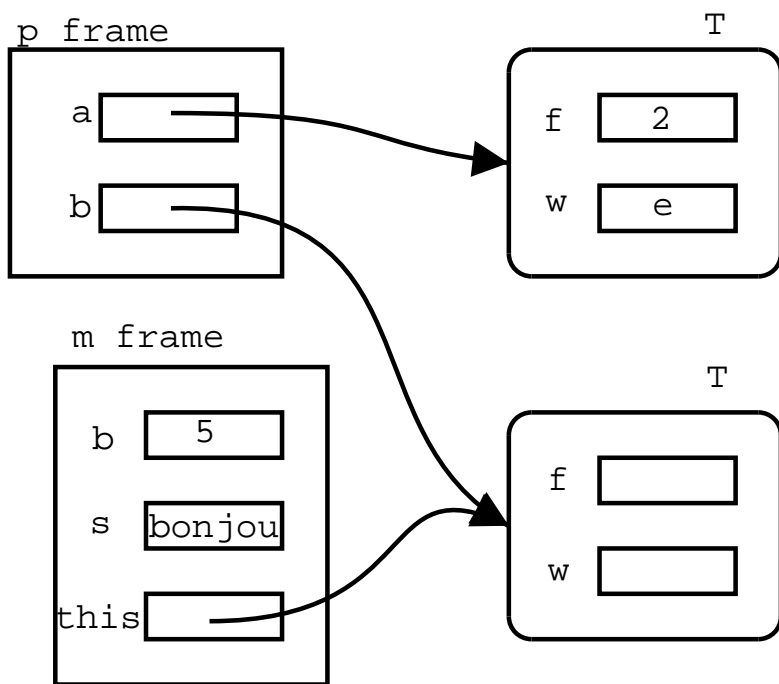
## Section 1: Multiple choice

Question 9 What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}
public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice





---

## Section 1: Multiple choice

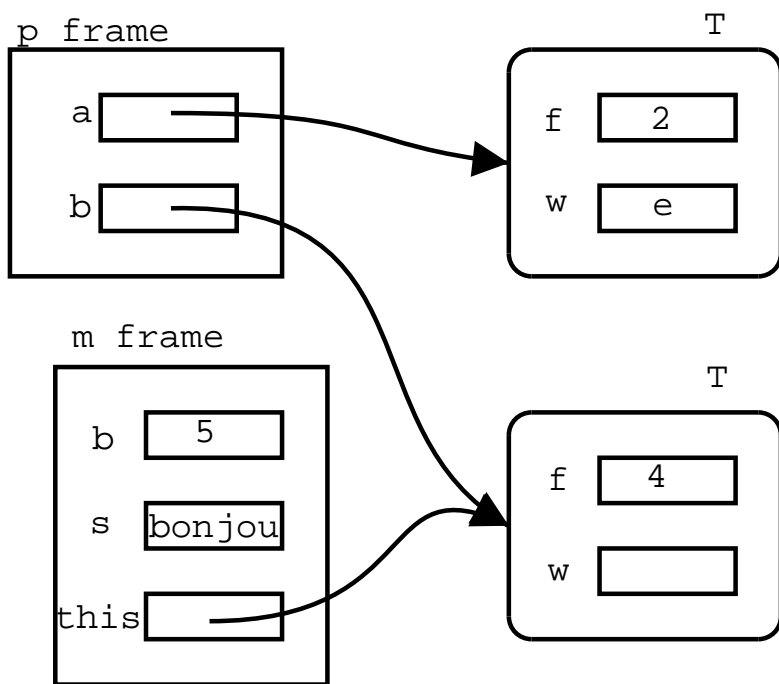
**Question 9** What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}

public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice



---

## Section 1: Multiple choice

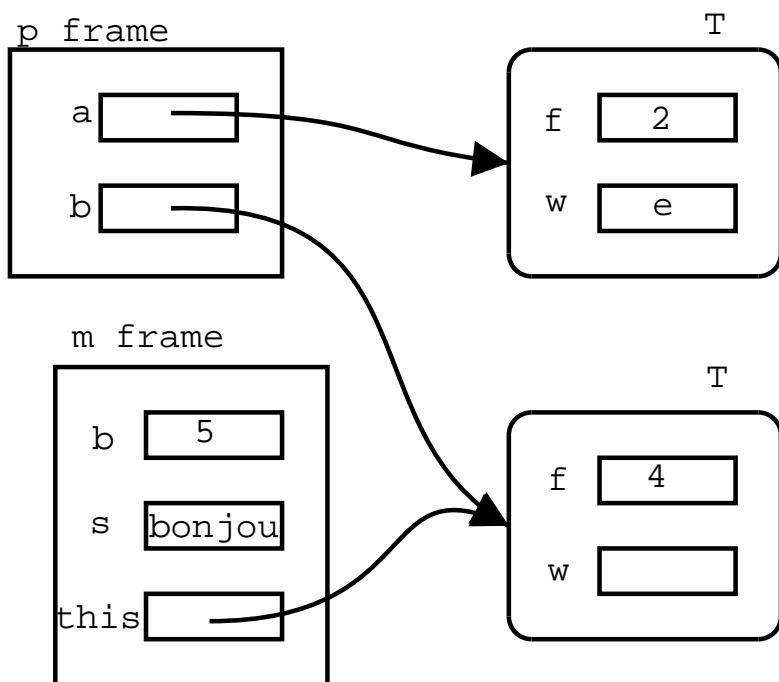
Question 9 What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}

public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice



---

## Section 1: Multiple choice

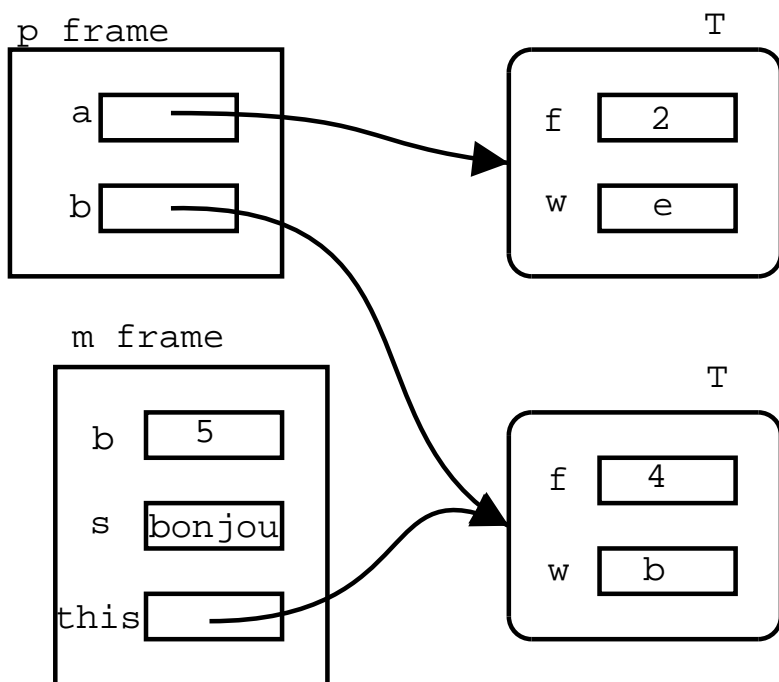
Question 9 What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}

public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice



---

## Section 1: Multiple choice

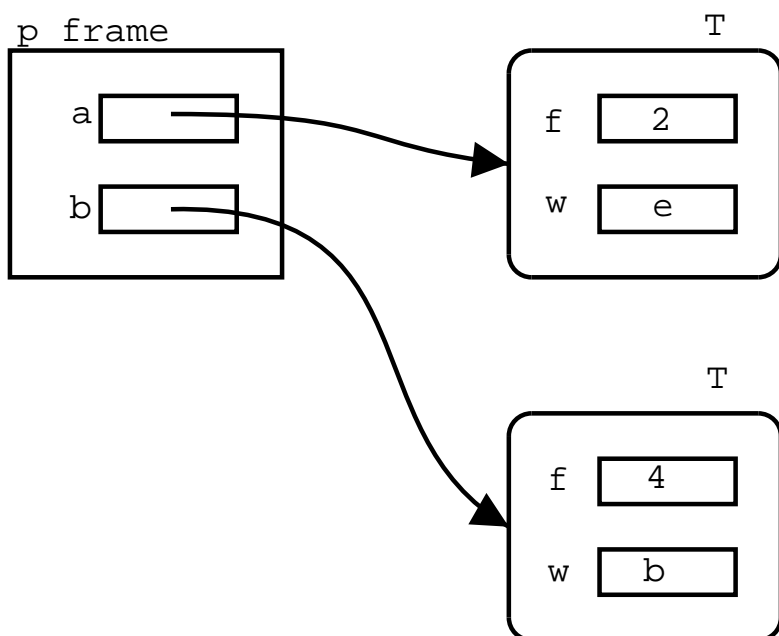
Question 9 What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}

public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice





---

## Section 1: Multiple choice

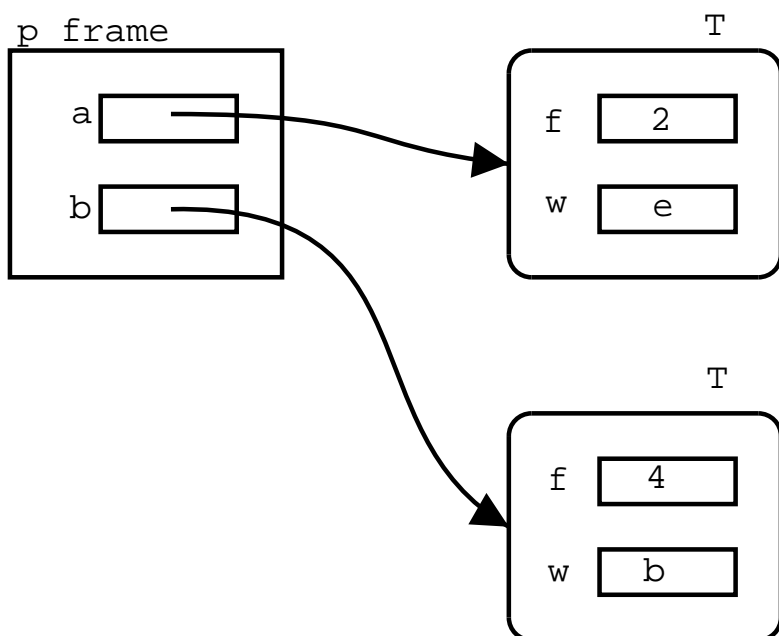
Question 9 What will this print if U.p() is called?

```
public class T {
    int f;
    char w;
    void m(int b, String s)
    {
        f = b - 1;
        if (f > 3) { w = s.charAt(0); }
        else { w = s.charAt(1); }
    }
}

public class U {
    static void p()
    {
        T a = new T(), b = new T();
        a.m(3, "hello");
        b.m(5, "bonjour");
        System.out.println(a.w + ", " + b.w);
    }
}
```

---

## Section 1: Multiple choice



---

## Section 1: Multiple choice

**Question 10** What does the following program fragment compute? (I.e. what is the output  $s$  in terms of the input  $n$ ?)

```
int n = Math.abs(Keyboard.readInt());
int i, c, s;
i = 0;
c = 0;
s = 0;
while (c < n) {
    if (i % 2 == 1) {
        s = i + s;
        c = 1 + c;
    }
    i = i + 1;
}
System.out.println(s);
```

---

## Section 1: Multiple choice

- a)  $s$  is the sum of the first  $n$  odd numbers.
- b)  $s$  is the sum of the first  $n$  even numbers
- c)  $s$  is the sum of the odd numbers smaller than  $n$ .
- d)  $s$  is the sum of the even numbers smaller than  $n$ .
- e) The program will not terminate

---

## Section 1: Multiple choice

- a)  $s$  is the sum of the first  $n$  odd numbers.
- b)  $s$  is the sum of the first  $n$  even numbers
- c)  $s$  is the sum of the odd numbers smaller than  $n$ .
- d)  $s$  is the sum of the even numbers smaller than  $n$ .
- e) The program will not terminate

Answer: a)

---

## Section 2: Short problems

**Question 1** Write a legal Java expression for each of the following conditions. You may assume that all variables are integers.

- a)  $x$  is less than 3 or  $y$  is less than 3 but not both.
- b)  $\alpha$  and  $\beta$  both have the same sign (both are positive or both are negative.) (Note: consider 0 to be positive.)
- c)  $x$  is not equal to either -1 or +1.

---

## Section 2: Short problems

**Question 1** Write a legal Java expression for each of the following conditions. You may assume that all variables are integers.

- a)  $x$  is less than 3 or  $y$  is less than 3 but not both.
- b)  $\alpha$  and  $\beta$  both have the same sign (both are positive or both are negative.) (Note: consider 0 to be positive.)
- c)  $x$  is not equal to either -1 or +1.

**Answer:**

a)  $(x < 3 \parallel y < 3) \&\& !(x < 3 \&\& y < 3)$

b)  $(\alpha \geq 0 \&\& \beta \geq 0) \parallel (\alpha < 0 \&\& \beta < 0)$

c)  $x \neq -1 \&\& x \neq +1$

---

## Section 2: Short problems

**Question 2** The following complete program is supposed to compute and print out the product of two numbers (5 and 8). It contains one (major) syntax error and one logic error. Correct both.

```
public class Product {
    int x = 5, y = 8, p = 0;
    while (x > 0) {
        x = x - 1;
        p = p + y;
    }
    System.out.print(x + " times " + y
                    + " equals " + p);
}
```



---

## Section 2: Short problems

**Answer:** Syntactic error: statements must go inside a method! Logic error: it prints "0 times 8 equals 40". A possible way to fix both:

```
public class Product {
    static void compute_product()
    {
        int x = 5, y = 8, p = 0, temp;
        temp = x;
        while (temp > 0) {
            temp = temp - 1;
            p = p + y;
        }
        System.out.print(x + " times " + y
            + " equals " + p);
    }
}
```

---

## Section 2: Short problems

**Answer:** Syntactic error: statements must go inside a method! Logic error: it prints "0 times 8 equals 40". A possible way to fix both:

```
public class Product {
    static void compute_product()
    {
        int x = 5, y = 8, p = 0, temp;
        temp = x;
        while (x > 0) {
            x = x - 1;
            p = p + y;
        }
        System.out.print(temp + " times " + y
            + " equals " + p);
    }
}
```

---

## Section 2: Short problems

**Question 3** What is wrong with this class? Point out the error and give two possible ways of fixing it:

```
public class CartoonCharacter {
    String name;
    static void work()
    {
        name = "Garfield";
        if (name.equals("Hobbes"))
            System.out.println("I'm sleeping");
        else
            System.out.println("I'm eating");
    }
}
```

---

## Section 2: Short problems

**Question 3** What is wrong with this class? Point out the error and give two possible ways of fixing it:

```
public class CartoonCharacter {
    String name;
    static void work()
    {
        name = "Garfield";
        if (name.equals("Hobbes"))
            System.out.println("I'm sleeping");
        else
            System.out.println("I'm eating");
    }
}
```

**Answer:** the variable `name` is non-static, and the static method `work` attempts to access `name`, but a static method cannot do it because static methods cannot affect objects directly.

---

## Section 2: Short problems

**Solution 1** Make name static:

```
public class CartoonCharacter {
    static String name;
    static void work()
    {
        name = "Garfield";
        if (name.equals("Hobbes"))
            System.out.println("I'm sleeping");
        else
            System.out.println("I'm eating");
    }
}
```

---

## Section 2: Short problems

**Solution 2** Make work non-static:

```
public class CartoonCharacter {
    String name;
    void work()
    {
        name = "Garfield";
        if (name.equals("Hobbes"))
            System.out.println("I'm sleeping");
        else
            System.out.println("I'm eating");
    }
}
```

---

## Section 2: Short problems

**Solution 3** Explicitly create an object from within the work method:

```
public class CartoonCharacter {
    String name;
    static void work()
    {
        CartoonCharacter g = new CartoonCharacter();
        g.name = "Garfield";
        if (g.name.equals("Hobbes"))
            System.out.println("I'm sleeping");
        else
            System.out.println("I'm eating");
    }
}
```

---

## Section 2: Short problems

**Question 4** Write down the printout of the following program. Does it terminate? If it does not terminate, just write the first 10 lines of printout, and state why it does not terminate. Otherwise, write the complete printout. State explicitly whether it terminates or not.

```
int p, q;
p = 7;
q = 2;
while (p - q >= 0) {
    if (p % 3 == 0) {
        p = p - 4;
        q = q + 1;
    }
    else {
        p = p + 1;
        q = q - 1;
    }
    System.out.println(p + “,” + q);
}
System.out.println(p + “,” + q);
```



---

## Section 2: Short problems

Answer: It terminates. The printout is:

8,1

9,0

5,1

6,0

2,1

3,0

-1,1

-1,1

---

## Section 3: Programming

**Question 1** Write a method which takes a positive integer as input, and determines whether the number is a “nacho” number or not. A “nacho” number is an odd “triangular” number. A “triangular” number is a number which is the sum of the first  $n$  positive integers. For example, the first 6 triangular numbers are 1, 3, 6, 10, 15, 21. (e.g.  $21 = 1 + 2 + 3 + 4 + 5 + 6$ ). Of these, only 1, 3, 15 and 21 are “nacho” numbers.

Before writing the actual Java code, write a **very brief** and general description of the algorithm to solve the problem (in English or French,) that makes your approach clear. Write this description as a sequence of clear steps (leave the details for the implementation.)

---

## Section 3: Programming

General Algorithm:

1. To find whether  $n$  is nacho we find out whether it is triangular and odd. If it is the case that both are satisfied, then  $n$  is nacho, otherwise it is not.

Algorithm to determine whether  $n$  is triangular:

1. Add all the natural numbers numbers from 0 until the sum is equal or larger than  $n$
2. If the sum is exactly  $n$ , then it is triangular. Otherwise it is not.

---

## Section 3: Programming

Refinement:

1. Set sum to 0
2. Set i to 0
3. While the sum is strictly less than n, repeat:
  - (a) add i to the sum
  - (b) increment i by 1
4. If sum is equal to n then return true
5. Otherwise return false

---

## Section 3: Programming

Implementation:

```
static boolean isNacho(int n)
{
    return isTriangular(n) && (n % 2 == 1);
}
static boolean isTriangular(int n)
{
    int i, sum;
    sum = 0;
    i = 0;
    while (sum < n) {
        sum = sum + i;
        i = i + 1;
    }
    return (sum == n);
}
```

---

## Section 3: Programming

Implementation:

```
static boolean isNacho(int n)
{
    if (isTriangular(n) && (n % 2 == 1))
        return true;
    return false
}
static boolean isTriangular(int n)
{
    int i, sum;
    sum = 0;
    i = 0;
    while (sum < n) {
        sum = sum + i;
        i = i + 1;
    }
    if (sum == n) return true;
    return false;
}
```

---

## Section 3: Programming

**Question 2** In this problem you will develop a small program that represents rectangles and performs some simple operations on them. To simplify the problem, rectangles will be considered to be “aligned” with the x and y axis, this is, the rectangles are not rotated, the base is parallel to the x axis, and the sides parallel to the y axis. A rectangle can be described by giving one of its corners, and its width and height.

---

## Section 3: Programming

**Part a)** (12 points) Write a class `Rectangle` that has:

- The coordinates of the top-left corner (`x,y`) (of type `integer`) and its `width` and `height` (integers too.)
- A constructor that initialises the instance variables (also called attributes or fields).
- An method called `area`, which returns the `area` of the rectangle.
- A method `int getX()` which returns the `x` coordinate of the top-left corner of the rectangle.
- A method `int getY()` which returns the `y` coordinate of the top-left corner of the rectangle.
- A method `int getWidth()` which returns the `width` of the rectangle.



- 
- A method `int getHeight()` which returns the height of the rectangle.
  - A method with signature `boolean contains(Rectangle r)`, which receives as input another rectangle `r`, and returns `true` if `r` is contained in (inside) the given rectangle and `false` otherwise. For example, if the rectangle to which this method is applied has top-left corner `(1,4)`, width 5 and height 3, and the rectangle `r` passed as input has top-left corner `(2,3)`, width 2 and height 1, then the result is `true`, but if `r` has top-left corner `(2,5)`, width 2 and height 2, the result is `false`. Note that this is not strict containment: if the rectangles are equal or they share a border (with `r` being inside,) then, the method should return `true`. For example, if the rectangle to which this operation is applied has top-left corner `(1,4)`, width 5 and height 3, and the rectangle `r` passed as input has top-left corner `(2,3)`, width 4 and height 2, then the result is also `true`.

---

## Section 3: Programming

**Part b)** (8 points) Write a program that asks the user to enter the information of two rectangles, say `a` and `b`, creates the two rectangle objects, prints the area of each, and prints whether `a` is contained in `b` and whether `b` is contained in `a`. You may use the `Keyboard` class.

---

## Section 3: Programming

```
public class Rectangle {
    int x, y, width, height;

    Rectangle(int x, int y, int width, int height)
    {
        this.x = x;
        this.y = y;
        this.width = width;
        this.height = height;
    }

    int area()
    {
        return width * height;
    }

    int getX()
    {
        return x;
    }
}
```

---

```
int getY()
{
    return y;
}

int getWidth()
{
    return width;
}

int getHeight()
{
    return height;
}

boolean contains(Rectangle r)
{
    return x <= r.x
        && y <= r.y
        && width >= r.width
        && height >= r.height;
}
}
```

---

## Section 3: Programming

Alternative:

```
boolean contains(Rectangle r)
{
    return this.x <= r.x
        && this.y <= r.y
        && this.width >= r.width
        && this.height >= r.height;
}
```

---

## Section 3: Programming

Alternative:

```
boolean contains(Rectangle r)
{
    if( this.x <= r.x
        && this.y <= r.y
        && this.width >= r.width
        && this.height >= r.height )
        return true;
    }
    return false;
}
```

---

## Section 3: Programming

```
import cs1.Keyboard;
public class RectangleTest
{
    public static void main(String[] args)
    {
        int x1, y1, w1, h1;
        int x2, y2, w2, h2;
        Rectangle a, b;

        System.out.println("Enter the top-left corner of the first rectangle: ");
        System.out.print("  Enter the x-coordinate: ");
        x1 = Keyboard.readInt();
        System.out.print("  Enter the y-coordinate: ");
        y1 = Keyboard.readInt();
        System.out.print("Enter the width: ");
        w1 = Keyboard.readInt();
        System.out.print("Enter the height: ");
        h1 = Keyboard.readInt();

        System.out.println("Enter the top-left corner of the second rectangle: ");
        System.out.print("  Enter the x-coordinate: ");
```

---

```
x2 = Keyboard.readInt();
System.out.print("  Enter the y-coordinate: ");
y2 = Keyboard.readInt();
System.out.print("Enter the width: ");
w2 = Keyboard.readInt();
System.out.print("Enter the height: ");
h2 = Keyboard.readInt();

a = new Rectangle(x1, y1, w1, h1);
b = new Rectangle(x2, y2, w2, h2);

System.out.println(a.area());
System.out.println(b.area());

if (a.contains(b))
    System.out.println("The first rectangle contains b");
if (b.contains(a))
    System.out.println("The second rectangle contains a");
}
}
```



---

## Static methods

- Static methods have no direct effect on objects
- When calling a static method, the frame does not have a “this” reference
- Therefore they cannot access any object attributes
- They can access only:
  - Parameters
  - Local variables
  - static variables (shared between all objects in the class)
- Non-static methods can access both static and non-static methods and variables
- Static methods cannot access non-static methods and non-static instance variables

---

## Objects as first class values

- Objects can be passed as parameters and returned as values

```
public class Rabbit {
    void jump() { ... }
}
public class Cage {
    void shake(Rabbit a)
    {
        a.jump();
    }
    Rabbit create()
    {
        return new Rabbit();
    }
}
```

...elsewhere...

```
Rabbit bugs = new Rabbit();
Cage c = new Cage();
c.shake(bugs);
Rabbit wester = c.create();
```

---

## Objects as first class values

- Objects can be attributes of other objects

```
public class Rabbit {
    void jump() { ... }
}
public class Cage {
    Rabbit my_rabbit;
    void put(Rabbit a)
    {
        my_rabbit = a;
    }
    Rabbit get()
    {
        return my_rabbit;
    }
}
```

...elsewhere...

```
Rabbit bugs = new Rabbit();
Cage c = new Cage();
c.put(bugs);
Rabbit wester = c.get();
```

---

## Recursion on other types

- Problem: given a string  $s$ , return the reverse of the string
- Analysis:
  - Notation:
    - \*  $\text{rev}(s)$  is the reverse of  $s$
    - \*  $s_i$  is the  $i$ -th character of  $s$
    - \*  $\text{len}(s)$  is the length of  $s$
    - \*  $\text{rest}(s)$  is the string  $s$  without its first character  $s_0$   
(i.e.  $\text{rest}(s) = s_1s_2\dots s_n$  where  $n = \text{len}(s) - 1$ )
  - Formal definition of reverse:

$$\text{rev}(s) = \begin{cases} "" & \text{if } s = "" \\ \text{rev}(\text{rest}(s)) + s_0 & \text{otherwise} \end{cases}$$

---

## Reverse (contd.)

- For example:

$$\begin{aligned}\text{rev}(\text{"abcd"}) &= \text{rev}(\text{"bcd"}) + 'a' \\ &= (\text{rev}(\text{"cd"}) + 'b') + 'a' \\ &= ((\text{rev}(\text{"d"}) + 'c') + 'b') + 'a' \\ &= (((\text{rev}(\text{""}) + 'd') + 'c') + 'b') + 'a' \\ &= (((\text{""} + 'd') + 'c') + 'b') + 'a' \\ &= ((\text{"d"} + 'c') + 'b') + 'a' \\ &= (\text{"dc"} + 'b') + 'a' \\ &= \text{"dcb"} + 'a' \\ &= \text{"dcba"}\end{aligned}$$

---

## Reverse (contd.)

```
public class MoreStringOperations {
    static String reverse(String s)
    {
        if (s.equals("")) {
            return "";
        }
        return reverse(rest(s))+s.charAt(0);
    }
    static String rest(String s)
    {
        String result = "";
        int i = 1;
        while (i < s.length()) {
            result = result + s.charAt(i);
            i++;
        }
        return result;
    }
}
```

---

## Double recursion

- Problem: Compute the  $n$ -th Fibonacci number
- Analysis: The Fibonacci sequence 1, 1, 2, 3, 5, 8, 13, 21, 34, ... is defined by:

$$fib(n) = \begin{cases} 1 & \text{if } n \leq 2 \\ fib(n-1) + fib(n-2) & \text{otherwise} \end{cases}$$

- Implementation:

```
static int fib(int n)
{
    if (n <= 2) {
        return 1;
    }
    return fib(n-1)+fib(n-2);
}
```

---

## Iteration vs recursion

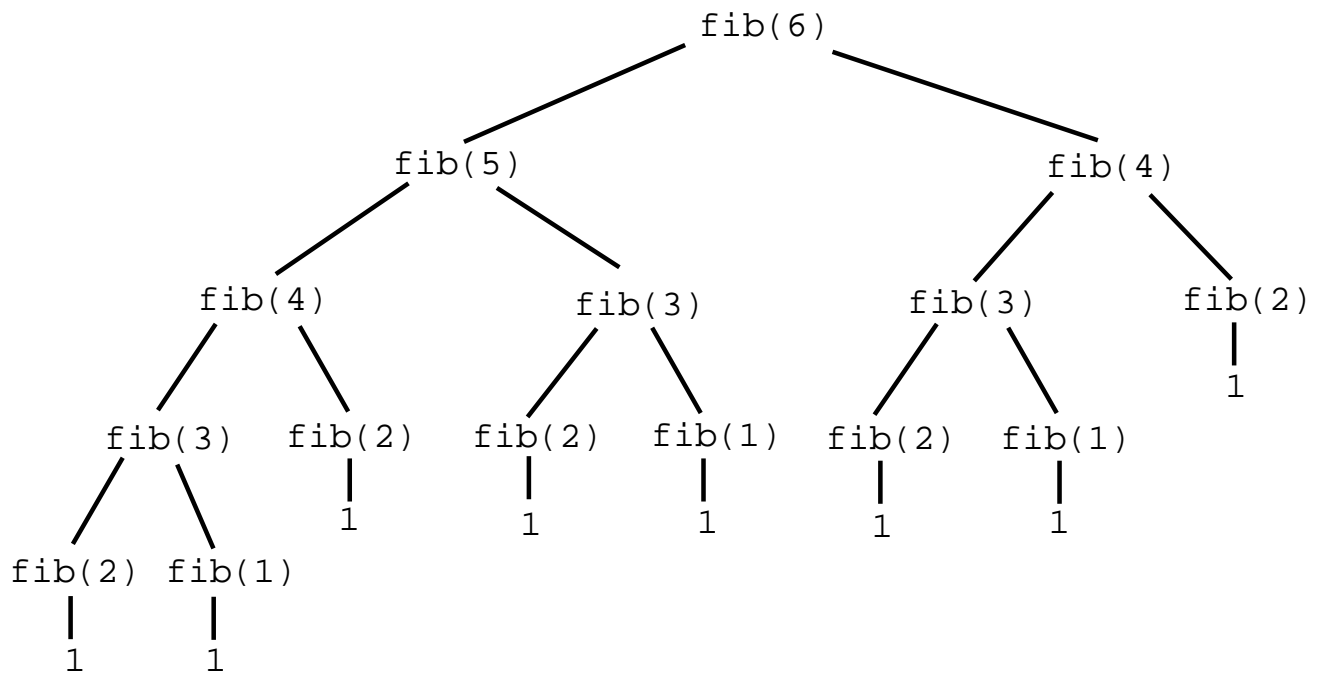
- Iterative solution to the Fibonacci problem:

```
static int fib(int n)
{
    int a, b, c, i;
    a = 1;
    b = 1;
    c = 1;
    i = 3;
    while (i <= n) {
        c = a + b;
        a = b;
        b = c;
        i++;
    }
    return c;
}
```



---

# Execution trees



---

## Recursion and termination

```
static void f(int n)
{
    System.out.println(n);
    f(n);
}
```

---

# Recursion and termination

$f(5)$

$f(5)$

$f(5)$

$f(5)$

.

.

.

---

## Recursion and termination

```
static int f(int n)
{
    System.out.println(n);
    return f(n) + 1;
}
```

---

## Recursion and termination

```
f(5)
return f(5) + 1
return f(5) + 1
return f(5) + 1
.
.
.
```

---

## Recursion and termination

```
static int f(int n)
{
    System.out.println(n);
    return f(n-1) + 1;
}
```

---

## Recursion and termination

```
f(5)
return f(4) + 1
return f(3) + 1
return f(2) + 1
return f(1) + 1
return f(0) + 1
return f(-1) + 1
.
.
.
```

---

## Recursion and termination

```
static int f(int n)
{
    if (n == 0) return 1;
    System.out.println(n);
    return f(n) + 1;
}
```



---

## Recursion and termination

```
f(5)
return f(5) + 1
return f(5) + 1
return f(5) + 1
return f(5) + 1
return f(5) + 1
return f(5) + 1
.
.
.
```

---

## Recursion and termination

```
static int f(int n)
{
    if (n == 0) return 1;
    System.out.println(n);
    return f(n-1) + 1;
}
```

---

## Recursion and termination

```
f(5)
return f(4) + 1
return f(3) + 1
return f(2) + 1
return f(1) + 1
return f(0) + 1
return 1 + 1
```

---

## Recursion and termination

```
f(5)
return f(4) + 1
return f(3) + 1
return f(2) + 1
return f(1) + 1
return 2
```

---

## Recursion and termination

```
f(5)
return f(4) + 1
return f(3) + 1
return f(2) + 1
return 2 + 1
```

---

## Recursion and termination

```
f(5)
return f(4) + 1
return f(3) + 1
return f(2) + 1
return 3
```

---

## Recursion and termination

```
f(5)
return f(4) + 1
return f(3) + 1
return 3 + 1
```

---

## Recursion and termination

```
f(5)
return f(4) + 1
return f(3) + 1
return 4
```



---

## Recursion and termination

```
f(5)
return f(4) + 1
return 4 + 1
```

---

# Recursion and termination

```
f(5)  
return 5 + 1
```

---

# Recursion and termination

```
f(5)  
return 6
```

---

## Recursion and termination

```
static int f(int n)
{
    if (n == 0) return 1;
    System.out.println(n);
    return f(n-2) + 1;
}
```

---

## Method overloading

- There can be several (static or not) methods with the same name...
- ...but the type or number of parameters must be different

---

## Example

```
public class A {
    void f(int x)
    {
        System.out.println("one: "+x)
    }
    void f(boolean x)
    {
        System.out.println("two: "+x)
    }
}
public class B {
    void g()
    {
        A u = new A();
        u.f(5);
        u.f(false);
    }
}
```

---

## Same for static methods

```
public class A {
    static void f(int x)
    {
        System.out.println("one: "+x)
    }
    static void f(boolean x)
    {
        System.out.println("two: "+x)
    }
}
public class B {
    void g()
    {
        A.f(5);
        A.f(false);
    }
}
```

---

The end