CS 132 Compiler Construction, Spring 2016
Instructor: Jens Palsberg
Multiple Choice Exam, Jun 9, 2016

ID	
Name	

This exam consists of 22 questions. Each question has four options, exactly one of which is correct, while the other three options are incorrect. For each question, you can check multiple options.

I will grade each question in the following way. If you check *none* of the options, you get 0 points. If you check all *four* options, you get 0 points.

Check one option. If you check *one* option, and that option is correct, you get 2 points. If you check *one* option, and that option is wrong, you get –0.667 points (yes, negative!).

Check two options. If you check *two* options, and one of those options is correct, you get 1 point. If you check *two* options, and both of them are wrong, you get –1 point (yes, negative!).

Check three options. If you check *three* options, and one of those options is correct, you get 0.415 points. If you check *three* options, and all three of them are wrong, you get –1.245 points (yes, negative!).

The maximum point total is $22 \times 2 = 44$ points. I will calculate a percentage based on the points in the following way:

$$\frac{\max(0, \text{point total})}{44} \times 100$$

Notice that if your point total is negative, you will get 0 percent.

Example
Consider the following Java program:

```
class E {
  int[] s;
  void mix(int[] d, int i) {
    int j;
    s = new int[13]; j = 5;
    while ( this.max(s.length,d.length) > j ) {
      if (j > i) { s[i] = d[j] + i; d[j] = d[j] + 1; }
      i = i + 2; j = j + 1;
    }
  }

  int max(int a, int b) {
    if ( b > a ) { return b; }
            else { return a; }
  }
}

class M {
  public static void main(String[] q) {
    E e = new E();
    e.mix(new int[17],0);
    System.out.println(e.s[2]);
  }
}
```

Question 1
Does the program type check?
- *a* ☐ Yes
- *b* ☐ No
- *c* ☐ The question cannot be answered based on the given information.
- *d* ☐ A Java type checker would throw an exception.

Question 2
What is the symboltable at the point where the type checker will try to type check the statement `j = 5;` in method `mix` in class `E` ?

a ☐ The following symboltable in which we search for identifiers from the bottom:

Id	Type
s	int[]
d	int[]
i	int
j	int

b ☐ The following symboltable in which we search for identifiers from the bottom:

Id	Type
d	int[]
i	int
j	int

c ☐ The following symboltable in which we search for identifiers from the bottom:

Id	Type
j	int

d ☐ The following symboltable in which we search for identifiers from the bottom:

Id	Type
E	void
s	int[]
d	int[]
i	int
j	int

Question 3

Consider the translation of the method `main` in class `M` to intermediate code in the style of Vapor. What is reasonable Vapor-style code for `e.mix(new int[17],0);` ? Assume that every word in newly allocated heap memory is automatically initialized to 0.

a ☐
```
t.0 = [e + 0]
t.1 = [t.0 + 0]
t.2 := HeapAllocZ 72
[t.2 + 0] = 17
call t.1(e, t.2, 0)
```

b ☐
```
t.0 = [e + 0]
t.1 = [t.0 + 0]
t.2 := HeapAllocZ 72
[t.2 + 0] = 17
call t.1(this, t.2, 0)
```

c ☐
```
t.0 = [e + 0]
t.1 = [t.0 + 0]
t.2 := HeapAllocZ 72
[t.2 + 0] = 17
call t.2(e, t.1, 0)
```

d ☐
```
t.0 = [e + 0]
t.1 = [t.0 + 0]
t.2 := HeapAllocZ 72
[t.2 + 0] = 17
call t.2(this, t.1, 0)
```

Question 4

Consider the translation of the method `mix` in class `E` to intermediate code in the style of Vapor. What is reasonable Vapor-style code for `s[i] = d[j] + i;` ? Assume that a static analysis in the compiler has found that array bounds checking is unnecessary.

a ☐
```
t.0 = 4 * (j + 1)
t.1 = d + t.0
t.2 = [t.1 + 0]
t.3 = t.2 + i
t.4 = 4 * (i + 1)
t.5 = [this + 4]
t.6 = t.5 + t.4
[t.6 + 0] = t.3
```

b ☐
```
t.0 = 4 * (j + 1)
t.1 = d + t.0
t.2 = [t.1 + 0]
t.3 = t.2 + i
t.4 = 4 * (i + 1)
t.5 = [this + 4]
t.6 = t.5 + t.4
[t.3 + 0] = t.6
```

c ☐
```
t.0 = 4 * (j + 1)
t.1 = d + t.0
t.2 = [t.1 + 0]
t.3 = t.2 + i
t.4 = 4 * (i + 1)
t.5 = [s + 4]
t.6 = t.5 + t.4
[t.6 + 0] = t.3
```

d ☐
```
t.0 = 4 * (j + 1)
t.1 = d + t.0
t.2 = [t.1 + 0]
t.3 = t.2 + i
t.4 = 4 * (i + 1)
t.5 = [s + 4]
t.6 = t.5 + t.4
[t.3 + 0] = t.6
```

Question 5

Consider the translation of the method `main` in class `M` to intermediate code in the style of Vapor. What is reasonable Vapor-style code for `E e = new E();` ? Assume that every word in newly allocated heap memory is automatically initialized to 0.

a ☐
```
e = HeapAllocZ 8
t.0 = Heapallocate 8
[e + 0] = t.0
[t.0 + 0] = E.mix
[t.0 + 4] = E.max
```

b ☐
```
e = HeapAllocZ 8
t.0 = Heapallocate 8
[t.0 + 0] = e
[t.0 + 4] = E.mix
[t.0 + 8] = E.max
```

c ☐
```
e = HeapAllocZ 8
t.0 = Heapallocate 12
[this + 0] = t.0
[t.0 + 0] = E.mix
[t.0 + 4] = E.max
```

d ☐
```
e = HeapAllocZ 8
t.0 = Heapallocate 12
[t.0 + 0] = e
[t.0 + 4] = E.mix
[t.0 + 8] = E.max
```

Question 6
Consider the translation of the method `max` in class `E` to intermediate code in the style of Vapor. Assume that we represent true by 1 and that we represent false by 0. What is reasonable Vapor-style code for the statement `if ( b > a ) { return b; } else { return a; }` ?

a ☐
```
      t.0 = LtS(a,b)
      if0 (t.0) goto :else
      return b;
      goto :end
else: return a;
end:
```

b ☐
```
      t.0 = LtS(a,b)
      if0 (t.0) goto :else
      return b;
      goto :else
else: return a;
end:
```

c ☐
```
      t.0 = LtS(b,a)
      if0 (t.0) goto :else
      return b;
      goto :end
else: return a;
end:
```

d ☐
```
      t.0 = LtS(b,a)
      if0 (t.0) goto :else
      return b;
      goto :else
else: return a;
end:
```

Question 7
Assume that the compiler does not map any temporaries, variables, or arguments to registers. What is the MIPS stack layout for the method `mix` in class `E` ? In each of the possible answers, the stack pointer points to bottom element.

a ☐ `d`
`i`
return address
`j`
saved registers

b ☐ `s`
`d`
`i`
return address
`j`
saved registers

c ☐ return address
`d`
`i`
`j`
saved registers

d ☐ return address
`s`
`d`
`i`
`j`
saved registers

Example
Consider the following program that uses the variables `b`, `i`, and `y`:

```
L1:  b = 1
L2:  if y > 10 then goto L4
L3:  y = y - b
L4:  b = b + 1
L5:  if b > 8 then goto L9
L6:  i = y + 2
L7:  b = y - 3
L8:  goto L3
L9:  b = b + i
L10: return b
```

Question 8
In the control-flow graph for the program, which variables are live along any of the edges to L1 ?

a ☐ `b`
b ☐ `y`
c ☐ `i, y`
d ☐ `b, i, y`

Question 9
In the control-flow graph for the program, which variables are live along any of the edges to L2 ?

a ☐ `b`
b ☐ `y`
c ☐ `b, y`
d ☐ `b, i, y`

Question 10
In the control-flow graph for the program, which variables are live along any of the edges to L3 ?

a ☐ `b, y`
b ☐ `b, i`
c ☐ `i, y`
d ☐ `b, i, y`

Question 11
In the control-flow graph for the program, which variables are live along any of the edges to L4 ?

a ☐ `b, y`
b ☐ `b, i`
c ☐ `i, y`
d ☐ `b, i, y`

Question 12
In the control-flow graph for the program, which variables are live along any of the edges to L5 ?

a ☐ `b, y`
b ☐ `b, i`
c ☐ `i, y`
d ☐ `b, i, y`

Question 13
In the control-flow graph for the program, which variables are live along any of the edges to L6 ?

a ☐ `i`
b ☐ `y`
c ☐ `i, y`
d ☐ `b, i, y`

Question 14
In the control-flow graph for the program, which variables are live along any of the edges to L7 ?
a ☐ `i`
b ☐ `y`
c ☐ `i, y`
d ☐ `b, i, y`

Question 15
In the control-flow graph for the program, which variables are live along any of the edges to L9 ?
a ☐ `b, y`
b ☐ `b, i`
c ☐ `i, y`
d ☐ `b, i, y`

Question 16
In the control-flow graph for the program, which variables are live along any of the edges to L10 ?
a ☐ `b`
b ☐ `i`
c ☐ `i, y`
d ☐ `b, i, y`

Example
The following is an entire procedure that uses six variables `c`, `d`, `e`, `f`, `g`, `h`. We assume that the two parameters `a1`, `a2` are assigned registers that are separate from the registers that we will use for the six variables.

```
     int m(int a1, int a2) {
L1:    c = a1 + a2
L2:    if (a1 > a2) goto L5
L3:    d = a1 + c
L4:    goto L8
L5:    e = a2 + c
L6:    f = a1 + e
L7:    d = a2 + f
L8:    g = d + 1
L9:    h = c + g
L10:   d = a1 + c
L11:   c = d + h
L12:   g = g + c
L13:   return g
     }
```

Question 17
Assume that we must assign one register to each of the six variables. What is the fewest number of registers that is needed for the six variables, *without* spilling?

a ☐ 3
b ☐ 4
c ☐ 5
d ☐ 6

Question 18
Assume that four registers `r1,r2,r3,r4` are available, and that we run the linear scan register allocation algorithm on the program. Which variables get spilled to memory?

a ☐ None
b ☐ `f`
c ☐ `g`
d ☐ `h`

Question 19
Assume that three registers `r1,r2,r3` are available, and that we run the linear scan register allocation algorithm on the program. Which variables get spilled to memory?

a ☐ None
b ☐ `f`
c ☐ `g`
d ☐ `h`

Question 20
Assume that two registers `r1,r2` are available, and that we run the linear scan register allocation algorithm on the program. Which variables get spilled to memory?

a ☐ `c`
b ☐ `g`
c ☐ `c,g`
d ☐ `g,h`

Example
The following program fragment uses four variables `c`, `d`, `g`, `h`:

```
c = d + h
g = g + c
```

Question 21
Assume that we have done register allocation for the procedure that contains the above program fragment, and that we have assigned

`c` to a memory location that is at offset 4 from register `fp`,
`d` to a memory location that is at offset 8 from register `fp`,
`g` to register `r1`, and
`h` to register `r2`.

Assume also that we have a spare register `v0`. What is correct code for the program fragment?

a ☐
```
v0 = [fp + 8]
v0 = v0 + r2
[fp + 4] = v0
r1 = r1 + v0
```

b ☐
```
v0 = [fp + 8]
v0 = v0 + r2
[fp + 4] = v0
v0 = [fp + 8]
r1 = r1 + v0
```

c ☐
```
v0 = [fp + 8]
v0 = v0 + r2
[fp + 4] = r1
r1 = r1 + v0
```

d ☐
```
v0 = [fp + 8]
v0 = v0 + r2
[fp + 4] = r1
v0 = [fp + 8]
r1 = r1 + v0
```

Question 22
Assume that we have done register allocation for the procedure that contains the above program fragment, and that we have assigned

`c` to register `r3`,
`d` to a memory location that is at offset 8 from register `fp`,
`g` to register `r1`, and
`h` to a memory location that is at offset 12 from register `fp`.

Assume also that we have a spare register `v0`. What is correct code for the program fragment?

a ☐
```
r3 = [fp + 8]
v0 = [fp + 12]
r3 = r1 + v0
r1 = r1 + r3
```

b ☐
```
v0 = [fp + 8]
r3 = [fp + 12]
r3 = r3 + v0
r1 = r1 + r3
```

c ☐
```
r3 = [fp + 8]
v0 = [fp + 12]
v0 = r1 + v0
r1 = r1 + r3
```

d ☐
```
v0 = [fp + 8]
r3 = [fp + 12]
v0 = r3 + v0
r1 = r1 + r3
```