

# Fall 2016 Programming Languages Qualifying Exam

This is a closed book test.

Correct, clear and precise answers receive full marks

Please start a new page for each question.

# Fall 2016 Programming Languages Qualifying Exam

## 1. *Static/Dynamic memory (25 pts)*

Consider the following Java program. Provide the output of the code and explain why the output is created as such. Explain where the variables **X**, **Y**, and **i** and their values are stored in memory (which memory segment), and how memory is (re)allocated for the assignment statements in the lines marked **A**, **B**, **C**, and **D**

```
class example {
    static StringBuffer Y = new StringBuffer(40); // A

    public static void printme(int k) {
        String X="a"; // B
        X=X+k; // C
        Y.append(X); // D
        System.out.println(X + "," + Y);
    }

    public static void main(String args[]) {
        for (int i=0; i<4; i++)
            printme(i);
    }
}
```

**a0,a0**  
**a1,a0a1**  
**a2,a0a1a2**  
**a3,a0a1a2a3**

**Y - Static, Data segment**  
**X - dynamic, stack segment**  
**i - dynamic, stack segment**

**A - Y is in the data segment pointing to Buffer in the heap**  
**B - X is on the stack referencing "a" is in the data segment**  
**C - X is on the stack referencing a new location on the heap**  
**D - Y is on data segment, pointing to a new location on the heap**

## Fall 2016 Programming Languages Qualifying Exam

### 2. Inheritance - What is the output of the following Java code (20 pts)

```
class Cass {
    public void method1() {
        System.out.print("cass 1 ");
    }
    public void method2() {
        System.out.print("cass 2 ");
    }
    public String toString() {
        return "cass";
    }
} // of Cass

class John extends Cass {
    public void method2() {
        method1();
        System.out.print("john 2 ");
    }
    public String toString() {
        return "john";
    }
} // of John

class Denny extends John {
    public void method1() {
        System.out.print("denny 1 ");
    }
    public String toString() {
        return "denny " + super.toString();
    }
} // of Denny

class tester {
    public static void main(String args[]) {
        Cass[] elements = {new Cass(),
                           new John(),
                           new JDenny()};
        for (int i = 0; i < elements.length; i++) {
            elements[i].method1();
            System.out.println();
            elements[i].method2();
            System.out.println();
            System.out.println(elements[i]);
            System.out.println();
        }
    }
} // of tester
```

# Fall 2016 Programming Languages Qualifying Exam

***cass 1***  
***cass 2***  
***cass***

***cass 1***  
***cass 1 john 2***  
***john***

***denny 1***  
***denny 1 john 2***  
***denny john***

## Fall 2016 Programming Languages Qualifying Exam

### 3. Grammars (20pts)

Rewrite the following arithmetic grammar in (E)BNF so that the new grammar implements the correct associativity and precedence rules of arithmetic (note that  $E^{**}E$  is exponentiation which is right to left).

$$E \rightarrow E + E \mid E - E \mid E * E \mid E^{**} E \mid E / E \mid ( E ) \mid \text{variable} \mid \text{num}$$

```
<E> ::=
    <E> + <term>
    | <E> - <term>
    | <term>

<term> ::=
    <term> * <factor>
    | <term> / <factor>
    | <factor>

<factor> ::=
    <element> ** <factor>
    | <element>

<element> ::=
    ( <E> )
    | variable
    | num
```

## Fall 2016 Programming Languages Qualifying Exam

### 4. Parameter Passing: Consider the following program (15 pts)

```
begin
  integer n;

  procedure p(j: integer)
  begin
    j := j+n;
    n := 2*n+j;
    print(n);
    print(j);
  end; // of procedure p

  n := 10;
  p(n);
  print(n);
end;
```

(a) What is the output when j is passed by value?

**40 20 40**

(b) What is the output when j is passed by value result?

**40 20 20**

(c) What is the output when j is passed by reference?

**60 60 60**

## Fall 2016 Programming Languages Qualifying Exam

5. **PYTHON - Consider the following python code. What is the output. Define the function "mystery()" (20pts)**

```
def suc (x):
    return x + 1

def mystery(k):
    l = list(range(2,k))
    for i in range(2,k):
        l = filter(lambda x: ( x==i ) or ( x%i !=0 ), l)
    return l

foo = [2, 18, 9, 22, 17, 24, 8, 12, 27]

print map(lambda x: suc (x), foo)
print filter(lambda x: x % 3 == 0, foo)

print mystery(50)
```

**[3, 19, 10, 23, 18, 25, 9, 13, 28]**

**[18, 9, 24, 12, 27]**

**[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]**

***mystery(k) returns a list of prime numbers upto and including k.***