CSC1003 Midterm Review

Introduction to Computer Science and Java Programming

孙超逸

计算机协会 (Computer @nd Comity)

Oct 18 2025

Compile

javac/java

```
javac [options] [source files]
java [options] classname [args]
```

.java $\xrightarrow{\text{Java Compiler (javac)}}$ .class JVM executable

`args/Scanner`

- `args`: Static input - provided at program startup
- `Scanner`: Dynamic input - collected during program execution

Quiz

```java
public class quiz {
    public static void main(String[] args) {
        String output = "";
        for (int i = 0; i < args.length; i++) {
            if (args[i].length() > 3) {
                output += args[i].charAt(3);
            }
            else {
                output += args[i];
            }
        }
        System.out.println(output);
    }
}
```

```
java quiz apple banana cat dog CSC1003
```

Primitive Type

- `byte` $[-2^7, 2^7 - 1]$
- `short` $[-2^{15}, 2^{15} - 1]$
- `int` $[-2^{31}, 2^{31} - 1]$
- `long` $[-2^{63}, 2^{63} - 1]$
- `float` 32-bit floating point
- `double` 64-bit floating point
- `char` 16-bit Unicode character
- `boolean` true/false values

Reference Types

- Classes (e.g., `String`)
- Interfaces
- Arrays

```
1 String s = "CSC1003";
```

Reference Types

- Classes (e.g., `String`)
- Interfaces
- Arrays

```
1 String s = "CSC1003";
```

```
1 String s = new String("abc");
2 String s1 = String.valueOf(100);
3 String s2 = Double.toString(3.14);
```

Quiz

```
1  public class quiz {
2      public static void main(String[] args) {
3          String a = 1 + 2 + "3";
4          String b = 1 + "2" + 3;
5          String c = "1" + 2 + 3;
6          System.out.println (a + b + c);
7      }
8  }
```

Formatted Output

`System.out.printf ();`

Specifier	Purpose	Example
%s	String	`System.out.printf("%s",str);`
%d	Integer	`System.out.printf("%d",val);`
%c	Character	`System.out.printf("%c",ch);`
%f	Float (6)	`System.out.printf("%f",val);`
%.nf	n Decimal	`System.out.printf("%.1f",1.23);`

Note: Rounding Behavior

- `%.nf` automatically rounds to n decimal places
- Example: `%.1f` rounds 1.25 $\rightarrow$ 1.3, 1.24 $\rightarrow$ 1.2
- Uses standard half-up rounding (0.5 rounds up)

if Statement

Base Syntax

```
1 if (condition) {
2     //if the condition if true
3 }
4 else {
5     //otherwise
6 }
```

Ternary Operator

```
1 condition ? exprIfTrue : exprIfFalse
```

Quiz

```
1  public class quiz {
2      public static void main(String[] args) {
3          int a = 15, b = 25, c = 10;
4          int val = (a > b) ?
5            ((a > c) ? a : c) :
6            ((b > c) ? b : c);
7          System.out.println (val);
8      }
9  }
```

Note that the program is not actually finding the maximum of three numbers.

Base Syntax

```
1 while (condition){
2     //repeat if condition is true
3 }
```

```
1 do{
2     //execute at least once
3     //repeat if condition is true
4 } while (condition);
```

Note: The semicolon ; after the while condition is mandatory!

for Statement

Base Syntax

```
1 for (initialization; condition; update){
2     //repeat until condition is false
3 }
```

Quiz

```java
public class quiz {
    public static void main(String[] args) {
        long n = Long.parseLong(args[0]);
        for (int i = 2; i <= n / i; i++) {
            while (n % i == 0) {
                System.out.print(i + ",");
                n = n / i;
            }
        }
        if (n > 1) System.out.println(n);
        else System.out.println();
    }
}
```

```
java quiz 228
```

Fundamental Operations

Operation	Array Code
Default initialization	`arr = new int[1000];`
Declare and initialize	`int[] arr = new int[1000];`
Literal initialization	`int[] arr = {1,2,3};` `int[] arr = new int[] {1,2,3};`

`int[] arr = new int[3] {1,2,3};`

Array constants can only be used in initializers!

Comparison with Strings

Array	String
`arr.length`	`str.length()`
`element = arr[index]`	`char = str.charAt(index)`
Mutable	Immutable

Copy

```
1 int[] a = {1, 2, 3};
2 int[] b = a;
3 int[] c = a.clone ();
4 System.out.println (a == b);
5 System.out.println (a == c);
6 b[0] = 99;c[1] = 99;
7 System.out.println (a[0]);
8 System.out.println (a[1]);
```

Quiz

```
1  for (int i = 0; i < n / 2; i++) {
2      int x = n - i - 1;
3      for (int j = i; j < x; j++) {
4          int y = n - j - 1;
5          int t = a[i][j];
6          a[i][j] = a[y][i];
7          a[y][i] = a[x][y];
8          a[x][y] = a[j][x];
9          a[j][x] = t;
10     }
11 }
```

Method Overloading

- Same method name with different parameters is allowed
- Parameters must differ in type, number, or order
- Return type alone cannot differentiate overloaded methods
- Compiler selects appropriate method based on arguments

```
1 public static void print (int number) { }
2 public static void print (String text) { }
3 public static int count (int a) { }
4 public static int count (int a,int b) { }
5 public static int cmp (int a,int b) { }
6 public static int cmp (int b,int a) { }
7 public static int calc (int a,int b) { }
8 public static double calc (int a,int b) { }
```

continue/break/return

Statement	Effect
break	Exits the current loop or switch statement immediately
continue	Skips the current iteration and moves to the next one
return	Exits the method and optionally returns a value

```
1  public static int calc (int n) {
2      if (n <= 0) return -1;
3      int sum = 0;
4      for (int i = 1; i <= n; i++) {
5          if (i % 2 == 1) continue;
6          if (i * i > n) break;
7          sum += i;
8      }
9      return sum;
10 }
```

Practice

```
1  public class Q1 {
2      public static void main(String[] args) {
3          int s = 0;
4          for (int i = 0; i < args.length; i++)
5              s += Integer.parseInt(args[i]);
6          int a = s / args.length;
7          double b = s / args.length;
8          System.out.println(a);
9          System.out.printf("%.1f\n",b);
10     }
11 }
```

Questions

- What is the command to compile the program?
- What is the output when running: `java Q1 1 2 3 4`?

Practice

```java
public class Q2 {
    public static void main(String[] args) {
        int[] array = new int[args.length];
        int start = 0,end = args.length - 1;
        while (start < end) {
            int temp = Integer.parseInt(args[start]);
            array[start] = Integer.parseInt(args[end]);
            array[end] = temp;
            start++;end--;
        }
        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i]);
            if (i != array.length - 1) {
                System.out.print(",");
            }
        }
    }
}
```

Questions

- What is the output when running: `java Q2 1 2 3 4 5 > result.txt`?
- If the program should reverse the input numbers, point out the mistake and correct it.

Questions

- What is the output when running: `java Q2 1 2 3 4 5 > result.txt`?
- If the program should reverse the input numbers, point out the mistake and correct it.

Reverse the array

```
1 for (int i = 0; i < n / 2;++i) {
2     int temp = arr[i];
3     arr[i] = arr[n - i - 1];
4     arr[n - i - 1] = temp;
5 }
```

Practice

```java
public class Q3 {
    public static void main(String args[]) {
        int n = Integer.parseInt(args[0]);
        int sum = 0, temp = n;
        while (n > 0) {
            sum = sum * 10 + n % 10;
            n /= 10;
        }
        if (temp == sum) System.out.println("true");
        else System.out.println("false");
    }
}
```

Questions

- What is the output when running: `java Q3 121212`?
- What is the output when running: `java Q3 121121`?

Palindrome

```
1  public class Q3 {
2      public static void main(String args[]) {
3          int n = Integer.parseInt(args[0]);
4          int sum = 0;
5          while (n > sum) {
6              sum = sum * 10 + n % 10;
7              n /= 10;
8          }
9          if (____1____) System.out.println("true");
10         else System.out.println("false");
11     }
12 }
```

Questions

- Complete the missing code at line 9.
- Any errors in the code? ($n \in \mathbb{N}$)

Practice

```java
public class Q4 {
    public static void main(String[] args) {
        int x = Integer.parseInt(args[0]);
        int y = Integer.parseInt(args[1]);
        int g = 1;
        for (int i = 1; i <= x && i <= y; i++) {
            if ((x % i) == 0 && (y % i) == 0)
                g = i;
        }
        System.out.println(g);
    }
}
```

Questions

- What is the output when running: `java Q4 24 84`?

gcd / lcm

```java
public class Q4 {
    public static void main(String[] args) {
        int x = Integer.parseInt(args[0]);
        int y = Integer.parseInt(args[1]);
        int l = x * y;
        for (int i = x * y; i >= x && i >= y; i--) {
            if (i % x == 0 && i % y == 0)
                l = i;
        }
        System.out.println(l);
    }
}
```

Questions

- What is the output when running: `java Q4 24 84`?

Practice

```java
public class Q6 {
    public static void main(String[] args) {
        int[] a = new int[args.length];
        int cnt = 0;
        for (int i = 0; i < args.length; i++)
            a[i] = Integer.parseInt(args[i]);
        for (int i = 0; i < a.length; i++)
            for (int j = i + 1; j < a.length; j++)
                if (a[i] > a[j]) {
                    int temp = a[i]; a[i] = a[j]; a[j] = temp;
                    ++cnt;
                }
        for (int i = 0; i < a.length; i++)
            System.out.print((a[i] + cnt) + "␣");
    }
}
```

Question

What is the output when running: `java Q6 3 8 9 5 2 4 1`?

Sort

```java
for (int i = 0; i < a.length-1; i++) {
    int minIndex = i;
    for (int j = i+1; j < a.length; j++)
        if (a[j] < a[minIndex]) minIndex = j;
    int temp = a[minIndex];
    a[minIndex] = a[i];
    a[i] = temp;
}
```

```java
for (int i = 1; i < a.length; i++) {
    int key = a[i], j = i - 1;
    while (j >= 0 && a[j] > key) {
        a[j+1] = a[j];
        j--;
    }
    a[j+1] = key;
}
```

Practice

Consider a program that takes a positive integer n as input and prints the binary representation of n.

```
1  public class Q7 {
2      public static void main(String[] args) {
3          int n = Integer.parseInt(args[0]), power = 1;
4          while (power < n/2) power *= 2;
5          while (power > 0) {
6              if (n < power) System.out.print(0);
7              else {System.out.print(1); n -= power;}
8              power /= 2;
9          }
10     }
11 }
```

Question

The program has one mistake. Point out the line number and correct it.

Practice

```
1  public class Q8 {
2      public static void main(String[] args) {
3          int n = 1000;
4          int[] array = new int[n];
5          for (int i = 0; i < n; i++)
6              array[i] = (int) Math.round(Math.random() * 255);//[0,255] in random
7          int[] counts = new int[255];
8          for (int i = 0; i < n; i++) _____
9          for (int i = 0; i < counts.length; i++) //Outputs the frequencies.
10             System.out.println(i + ":" + counts[i]);
11     }
12 }
```

Question

- Point out the line number of the mistake and correct it.
- Complete the missing code at line 8 so the program correctly counts frequencies.

Practice

```java
public class Q9 {
    public static int f(int[] a, int n) {
        int i1 = 0, i2 = a.length, c = 0;
        while (true) {
            c++;
            int i = (i1 + i2) / 2;
            if (a[i] == n) return c;
            if (c > a.length) return -1;
            if (a[i] < n) i1 = i;
            if (a[i] > n) i2 = i;
        }
    }
    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        int[] array = new int[100];
        for (int i = 0; i < array.length; i++) array[i] = i;
        System.out.println("c=" + f(array, n));
    }
}
```

Question

- What is the output when running: `java Q9 31`?
- What is the output when running: `java Q9 128`?
- For input integers between 0 and 99, what is the largest output value of c?

Binary Search

```
1 public static void main(String[] args) {
2     int n = Integer.parseInt(args[0]);
3     int[] array = new int[n];
4     for (int i = 0; i < array.length; i++) array[i] = i;
5     int cnt = 0, _max = (int) Math.ceil(Math.log(n) / Math.log(2));
6     for (int i = 0; i < n; i++)
7         if (f(array,i) == _max) ++cnt;
8     System.out.println(cnt);
9 }
```

Binary Search

```java
public static void main(String[] args) {
    int n = Integer.parseInt(args[0]);
    int[] array = new int[n];
    for (int i = 0; i < array.length; i++) array[i] = i;
    int cnt = 0, _max = (int) Math.ceil(Math.log(n) / Math.log(2));
    for (int i = 0; i < n; i++)
        if (f(array,i) == _max) ++cnt;
    System.out.println(cnt);
}
```

Answer

Let $k = \lfloor \log_2(n) \rfloor$.

The number of midpoints discovered in at most k iterations is $2^0 + 2^1 + \cdots + 2^{k-1} = 2^k - 1$.

Thus the remaining indices (found first at iteration $k+1$) are $n - (2^k - 1) = n - 2^{\lfloor \log_2(n) \rfloor} + 1$.

- Ready to become a coding **MASTER**?
- Don't miss the school competition next semester!

Q & A

Thanks for listening!

Good Luck on Your Midterm Exams!