

CSIS 0911B Computer Concepts and Programming
for non-Engineering students (Java)

Assignment 3

Assigned 2:00pm Oct 30, 2000; dead 2:00pm Nov 20, 2000.
Assignment box: R5, lift lobby, 3/F Chow Yei Ching building

Submission: Write down your answers to the questions. Save your program in section 3 in a floppy diskette. Also print a copy of your program. Put all the above into an envelope and put it into assignment box R5. **Remember to write down your name, university ID and your curriculum** on the envelope. Send a mail to the tutor (cpng@csis.hku.hk) indicating that you have submitted an assignment.

Section 1: Short Questions

1. Fill in the blanks, so that the desired output is printed when a is an int type variable with value 20, and b is a double type variable with value 275000.

- (a) `Fmt.printf([], a); // Want "00020"`
- (b) `Fmt.printf([], b); // Want "HKD 275000.00"`
- (c) `Fmt.printf([], b); // Want " +2.750E+005kg"`

2. Try to trace the following program and see what will be printed.

```
public class BreakTest {
    public static void main(String [] args) {
        loop:
            for (int i = 0; i < 10; ++i) {
                switch (i) {
                    case 5:
                        System.out.print("Hello ");
                    case 7:
                        System.out.print("World!");
                    case 2:
                        break;
                    case 6:
                    case 8:
                        System.out.println();
                        break loop;
                    case 3:
                        continue;
                    default:
                        System.out.println("Good morning!");
                }
                System.out.print(' ');
            }
        }
    }
```

3. Try to run the following program. Is the result what you expect? Why? What do you learn from this example?

```
/* Print the squares of numbers from 0.1 to 2.0 */
import chapman.io.*;
public class Loop {
    public static void main(String[] args) {
        double i;
        for (i=0.1; i<=2.0; i+=0.1)
            Fmt.printf("%4.2f\n", i*i);
    }
}
```

4. Explain what will happen when the following program is executed, and why.

```
public class SwapThings {
    public static void main(String[] args) {
        int a = 2, b = 3;
        swap(a, b);
        System.out.println("a = " + a + "; b= " + b);
    }
    public static void swap(int x, int y) {
        int t = x;
        x = y;
        y = t;
    }
}
```

5. Write a method `smallestFactor` that accept an integer argument, and return the smallest factor larger than 1 of this number. It should return the argument itself if it is a prime or it is 1, and return 0 if it is not positive.
6. By calling the method you've written in 5, write the main method of a program that read a number from the user and completely factorize it.
7. Compare the following two programs. What type of error they produces? Why we prefer the style on the right than the style on the left?

<pre>public class Err1 { public static int getChar() { char ch; ... // Read a character here if (Character.isDigit(ch)) return 1; else return 0; } public static void main (String[] args) { if (getChar() == 2) { ... } } }</pre>	<pre>public class Err2 { final static int FOUND_LETTER = 0; final static int FOUND_DIGIT = 1; public static int getChar() { char ch; ... // Read a character here if (Character.isDigit(ch)) return FOUND_DIGIT; else return FOUND_LETTER; } public static void main (String[] args) { if (getChar() == FONUD_DIGIT) { ... } } }</pre>
--	--

8. The following may be the most “beautiful” program you’ve seen artistically, but it is very difficult to understand. Improve the program by making proper line breaking, variable names, method names and program names; turning all class variables into function arguments, and also remove redundant expressions. Understand the program, and write comments for it.

```
import chapman          .io.*;public          class a {public
static void f()          {System .out.          print( '#');for
(j=2;j <w;++j)          if(i==j|| i+j==w+1)      System. out.print
('*');else System.      out.print(' ');System.out.    println ('#');}public
static void main        (String[] args){StdIn  t=new StdIn ();w=t.
readInt ();for(i=      0;i<w;++i)System.out.    print( '#');System.
out.println();for(i    =2;i<w;++ i)f();for(i=      0;i<w;++i)System.out.
print('#');System.      out.println ("");}static  int i,j,w;}
```

9. In general, we can say something like “the scope of this Java local variable is from this Java statement until this Java statement”. But we cannot say something like “the duration of this Java local variable is from this Java statement until this Java statement”. Why?

10. Write Java statements to declare an array of 100 double type values, allocate memory for it, and fill it with 0.0, 0.1, ..., 9.9.

Section 2: Program Debugging

The following program contains a method that attempts to make the program to make changes looks better. It also contains a main method to test it. However, there are some errors in the program. Fix all errors in the following program, without modifying the intuitive meaning, so that it can be executed and it makes sense:

```
import chapman.io.*;
public class MakeChanges {
    public static void main(String[] args) {
        StdIn in = new StdIn();
        System.out.print("The amount to change? (<=0 exits) ");
        int amount = in.readInt();
        while (amount > 0) {
            moneyReturn();
            System.out.print("The amount to change? (<=0 exits) ");
            amount = in.readInt();
        }
    }
    /* Make changes for AMOUNT cents */
    public static void moneyReturn() {
        amount = makeChangeWith("1000");
        amount = makeChangeWith("500");
        amount = makeChangeWith("200");
        amount = makeChangeWith("100");
        amount = makeChangeWith("50");
        amount = makeChangeWith("20");
        amount = makeChangeWith("10");
    }
    /* Make change for AMOUNT cents using only COIN cents coin */
    public static int makeChangeWith(int coin) {
        while (remains >= coin) {
            remains -= coin;
            System.out.println("Drop $" + coin/100.0 + " coin");
        }
    }
    public static int amount = 0;
}
```

After fixing the whole program, re-introduce each error into the program one-by-one, and compile the program each time an error is re-introduced. Write down the error message that the compiler outputs, and the intuitive meaning of the error message.

Section 3: Program Development

Minesweeper is a game that is played on a rectangular playing field, which consists of a grid of cells. The playing field contains a certain number of invisible **mines**. The player visits cells in the field. Whenever the player visit a cell without a mine, he know how many of the (at most 8) cells surrounding him contains a mine. With this information, the player tries to visit all cells without a mine. The player loses if he visits a cell with a mine, and wins if he visits all cells which does not contain a mine.

Write a program to allow the user to play minesweeper. The program starts by asking the width (at most 40) and height (at most 20) of the playing field, followed by the number of mines there. It then generates a field of that dimension, and place random mines into the field. Then it repeatedly displays the current field and ask the user to visit a cell. It determines erratic conditions like a negative number of cells, or the user want to visit a cell outside the field. It also detects the end of the game, when the full field is revealed to the player. You may use any intuitive way to display the playing field, although the following example may serve as a guideline.

As an option, your program may automatically visit all the cells surrounding a visited cell that is known to be adjacent to no mine. If you can think of other enhancements of the program, you are encouraged to add that into your program as well.

Hint: `Math.random()` create a random number of at least 0 but always less than 1. If you want a random number from 0 to n-1, you can use `(int)(Math.random() * n)`.

(Note: due to a problem in the `readChar` method of `chapman.io.StdIn`, the program will generate a run-time error if it asks for a character and the user just type enter. You may ignore this error.)

Example input/output (with the optional feature):

```
Width? 5
Height? 5
Number of mines? 3
  ABCDE
  1
  2
  3
  4
  5
```

```
Move: x? a
      : y? 1
Visited 1/ 22
```

```
  ABCDE
  1 1
  2
  3
  4
  5
```

```
Move: x? a
      : y? 2
Visited 2/ 22
```

```
  ABCDE
  1 1
  2 2
  3
  4
  5
```

```
Move: x? g
Invalid x. Please try again.
```

```
Move: x? e
      : y? 1
Visited 11/ 22
```

```
  ABCDE
  1 1 100
  2 2 100
  3  111
  4
  5
```

```
Move: x? a
      : y? 3
Oops... killed!
```

```
  ABCDE
  1 1 100
  2 2*100
  3 # 111
  4   *
  5
```