COMS W1004 MW 4:10-5:25 PM
Columbia University Spring 2014

Homework 2: Problem Set
February 17, 2014

Your CUNIX ID: ami2119

Your Last Name: Iwamizu

Your First Name: Akiko

Circle the range that includes your UNI:

Group 1 (aa3473-amb2368)

Group 2 (amg2300-bac2176)

Group 3 (bbcb2137-cyb2105)

Group 4 (db2941-erw2138)

Group 5 (es2680-ip2273)

Group 6 (it2217-jmf2211)

Group 7 (jmg2227-kgl2111)

Group 8 (kh2633-mc3834)

Group 9 (mcs2225-nam2182)

Group 10 (nay2109-rdy2104)

Group 11 (rhj2002-sec2161)

Group 12 (sfm2123-tla2119)

Group 13 (tlb2145-zn2116)

Chapter 4: R4.5, R4.7, R4.9, R4.10

R4.5) Write as a mathematical expression in Java

```
double s = s0 + v0 * t + (1.0/2.0) * g * t * t;

double G = 4 * Math.PI * Math.PI * Math.pow(a, 3) / (p * p * (m1 + m2));

double FV = PV * Math.pow((1 + (INT/100) ) , YRS);

double c = Math.sqrt(a * a + b * b – 2 * a * b * Math.cos(gamma));
```

R4.7) Solve where n = 17 and m = 18

a. n / 10 + n % 10 = (1) + (7) = **8**

b. n % 2 + m % 2 = (1) + (0) = **1**

c. (m + n) / 2 = (35) / 2 = **17**

d. (m + n) / 2.0 = (35) / 2.0 = 17.**5**

e. (int) (0.5 * (m + n)) = (int) (17.5) = **17**

f. (int) Math.round(0.5 * (m + n)) = (int) (18) = **18**

R4.9) Find at least five compile-time errors in the following program

1. Should define “in”
2. String [] args need to be in “main()”
3. Delete “;” after “main();”
4. Need to declare “x” and “y”
5. Enter quotes when using println, so println(“Please enter two numbers:”);
6. “printline” does not work, so use “println” instead

```
public class HasErrors
{
   public static void main();
    {
         System.out.println(Please enter two numbers:);
         x = in.readDouble();
         y = in.readDouble();
         System.out.printline(“The sum is ”  + x + y);
     }
 }
```

R4.10) Find three run-time errors in the following program

1. When identifying “in”, there should **not** be quotes in Scanner(“System.in”); for this program.
2. When printing out the sum, “x + y” combines the integers after printing them, whereas they wanted to print the sum of these two integers (ie. For x = 12 and y = 34, prints 1234 instead of 46).
3. In addition, “x” is initialized twice whereas “y” needed to be initialized. The result of this error prints out the value of “x” and the value of “y” which is 0 (ie. For x=24 and x= 45, prints 450).

```
public class HasErrors
{
   public static void main(String[] args)
    {
         int x = 0;
         int y = 0;
         Scanner in = new Scanner(“System.in”);
         System.out.print(“Please enter an integer:”);
         x = in.readInt();
         System.out.print(“Please enter another integer: ”);
         x = in.readInt();
         System.out.println(“The sum is ” + x + y);
    }
}
```

Chapter 5: R5.4, R5.11, R5.16 (notice the algorithm it asks about is in the PRACTICE EXERCISES section)

R5.4) What do these code fragments print?

a. **-1**

b. **1**

c. **1**

d. **Math.sqrt(2)**

R5.11) Determine whether two appointments overlap.

1. An appointment from 10 – 12 and 11 – 13. Let start1=10, end1=12, start2=11, and end2=13.
    - **Overlap**

If 10 > 11 //FALSE
    s = start1
Else
    s = 11
If 12 < 13 //TRUE
    e = 12
Else
    e = end2
If 11 < 12 //TRUE
    The appointments overlap.
Else
    The appointments don't overlap.

2. An appointment from 10 – 11 and 12 – 13. Let start1=10, end1=11, start2=12, and end2=13.
    - **No Overlap**

If 10 > 12 //FALSE
    s = start1
Else
    s = 12
If 11 < 13 //TRUE
    e = 11
Else
    e = end2
If 12 < 11 //FALSE
    The appointments overlap.
Else
    The appointments don't overlap.

R5.16) Develop a set of test cases for the algorithm in E5.14.

If month is 1, 2, or 3, season = “Winter”
Else if month is 4, 5, or 6, season = “Spring”
Else if month is 7, 8, or 9, season = “Summer”
Else if month is 10, 11, or 12, season = “Fall”
If month is divisible by 3 and day >= 21
    If season is “Winter”, season = “Spring”
    Else if season is “Spring”, season = “Summer”
    Else if season is “Summer”, season = “Fall”
    Else season = “Winter”

Test Case:	Month	Day	Expected Output	Comment
1	1	12	Winter	Boundary case
2	3	20	Winter	Boundary case
3	3	24	Spring	Boundary case
4	6	20	Spring	Boundary case
5	6	25	Summer	Boundary case
6	9	20	Summer	Boundary case
7	9	21	Fall	Boundary case
8	12	19	Fall	Boundary case
9	12	30	Winter	Boundary case
10	0	0	?	Receive error
11	1	40	Winter	Invalid input
12	24	3	?	Receive error
13	%	$	?	Receive error

Chapter 6: R6.1, R6.15, R6.18

R6.1) Write a while loop that prints

a. All squares less than n. For example, if n is 100, print 0 1 4 9 16 25 36 49 64 81.

```
import java.util.Scanner;
public class Squares
{
  public static void main(String[] args)
  {
        Scanner input = new Scanner(System.in);
        System.out.println(“Please enter an integer: ”);
        Int i = 0;
        Int n = 0;
        Int x = input.nextInt();

    while (i < x) {
    System.out.print(i + “ ”);
    n++;
    i = n*n;
    }
  }
}
```

b. All positive numbers that are divisible by 10 and less than n. For example, if n is 100, print 10 20 30 40 50 60 70 80 90.

```
import java.util.Scanner;
public class Divisible
{
  public static void main(String[] args)
  {
        Scanner input = new Scanner(System.in);
        System.out.println(“Please enter an integer: ”);
        int i = 10;
        int x = input.nextInt();

    while (i < x) {
    System.out.print(i  +  “ ”);
    i = i+10;
    }
  }
}
```

c. All powers of two less than n. For example, if n is 100, print 1 2 4 8 16 32 64.

```
import java.util.Scanner;
public class Divisible
{
  public static void main(String[] args)
  {
        Scanner input = new Scanner(System.in);
        System.out.println("Please enter an integer: ");
        int i = 1;
        int x = input.nextInt();

    while (i < x) {
    System.out.print(i  +  " ");
    i = i*2;
    }
   }
}
```

R6.15) Rewrite the for loop into a while loop

```
int s = 0;
int i = 1;
while (i <= 10)
  {
     s = s + i;
     i++;
  }
```

R6.18) What do the following loops print?

a. 2 4 7 11 16
b. 4 9 16
c. 10 7