

CIS 1068  
Program Design and Abstraction  
Fall 2016  
Midterm Exam 2

Name \_\_\_\_\_

Page	Points	Score
2	6	
3	15	
4	6	
5	26	
6	10	
7	10	
8	10	
9	17	
Total	100	

## 1. Program Traces (53 points, 60 minutes)

a. (6 points) Answer the questions below.

```
public class C1 {  
    public static void mystery (int x, boolean y) {  
        // POINT 1  
        x += 2;  
        y = !y;  
        // POINT 2  
    }  
    public static void main (String [] args) {  
        int x = 3;  
        boolean y = false;  
        mystery(x, y);  
        // POINT 3  
    }  
}
```

1. What are the values of x and y at POINT 1?

3, false

2. What are the values of x and y at POINT 2?

5, true

3. What are the values of x and y at POINT 3?

3, false

b. (6 points) What will be on the screen display after the program is executed?

```
public class C2 {  
    public static int mystery1 (int [] d) {  
        int c = 0;  
        for(int i=1; i<d.length; i++){  
            if (d[i]> d[0]) c++;  
        }  
        return c;  
    }  
  
    public static void main (String [] args) {  
        int [] d1 = {1, 2, 3, 4, 5};  
        int [] d2 = {5, 4, 3, 2, 1};  
        int [] d3 = {3, 2, 4, 5, 1};  
  
        System.out.println(mystery1(d1));  
        System.out.println(mystery1(d2));  
        System.out.println(mystery1(d3));  
    }  
}
```

4

0

2

c. (9 points) What will be on the screen display after the program is executed?

```
public class C3 {
public static int mystery2 (int [ ] d) {
int c = d.length;
for(int i=0; i<d.length-2;){
if (d[i++]<c) c=d[i++];
}
return c;
}

public static void main (String [] args) {
int [ ] d1 = {6, 7, 8, 9, 10};
int [ ] d2 = {5, 4, 3, 2, 1};
int [ ] d3 = {3, 2, 4, 5, 1};

System.out.println(mystery2(d1));
System.out.println(mystery2(d2));
System.out.println(mystery2(d3));
}
}
```

Array d1: each if d[i]: d[0], d[1], d[2] is false, so c = 5 until i = 3! Screen display: **5**

Array d2: d[1] < c, then i=2; c=d[2]=3. After that, i=3 and stop the loop! Screen display: **3**

Array d3: d[0] < c, then i = 1; c = d[1] = 2. After that, i=2. If d[2]=4 < c = 2 is false. Go to i=3 and stop!  
Screen display: **2**

d. (6 points) Answer the questions below.

```
public class C4{
public static void mystery1 (int [ ] d) {
d = null;
}
public static void mystery2 (int [ ] d) {
d = new int[d.length];
}
public static void mystery3 (int [ ] d) {
for( int i = 0; i< d.length; i++) d[i] = 0;
}

public static void main (String [] args) {
int [ ] d1 = {1, 2, 3};
mystery1(d1);
// POINT 1
int [ ] d2 = {1, 2, 3};
mystery2(d2);
// POINT 2
int [ ] d3 = {1, 2, 3};
mystery3(d3);
// POINT 3
}
}
```

1) Draw what d1 looks like at POINT 1.

{1, 2, 3}

2) Draw what d2 looks like at POINT 2.

{1, 2, 3}

3) Draw what d3 looks like at POINT 3.

{0, 0, 0}

e. (26 points) For each snippet of Java code on the left, write down the value of the variables after the code is finished executing.

Code	Class definition	Value of variable at end of Code in LEFT COLUMN
Mystery1 m = new Mystery1();	public class Mystery1 { private double a=1.5; public Mystery1() { a = 2.5; } }	m = {2.5}
Mystery2 m = new Mystery2(); m.a = 1.0; m.update();	public class Mystery2 { public double a; public void update() { a += 2.5; } }	m = {3.5}
Mystery3 m1 = new Mystery3(); int x1 = m1.getVal(); Mystery3 m2 = new Mystery3(); int x2 = m2.getVal(); x2 = m2.getVal();	public class Mystery3 { private int a = 2, b = 3; public int getVal() { a = a + b; return a; } }	x1 = 5 m1 = {5,3} x2 = 8 m2 = {8,3}
Mystery4 m1 = new Mystery4(); int x1 = m1.getVal(); Mystery4 m2 = new Mystery4(); int x2 = m2.getVal(); x2 = m2.getVal();	public class Mystery4 { private static int a = 2, b = 3; public int getVal() { a = a + b; return a; } }	x1 = 5 m1 = {} x2 = 11 m2 = {}
Mystery5 m1 = new Mystery5(); String str1 = m1.getStr(0);  Mystery5 m2 = new Mystery5(); String str2 = m2.getStr(2);	public class Mystery5 { private String s = "ab"; int n = 3; public String getStr (int num){ n = num; for(int i=0;i<n;i++){ s = s + n; } return s; } }	m1 = {"ba", 3} str1 = "ba" m2 = {"ba22", 3} str2 = "ba22"
Mystery6 m1 = new Mystery6(1,2);  Mystery6 m2 = new Mystery6(2,1);	public class Mystery6 { public int [] a = null; public Mystery6 (int len, int val){ a = new int[len]; for(int i=0;i<len; i++){ a[i] = val; } } }	m1 = {[2]} m2 = {[1,1]}

## 2) Writing short programs (30 points, 40 minutes)

a. (10 points) Write a static method named `enoughTimeForLunch` that accepts four integers `hour1`, `minute1`, `hour2`, and `minute2` as parameters. Each pair of parameters represents a time on the 24-hour clock (for example, 1:36 PM would be represented as 13 and 36). The method should return true if the gap between the two times is long enough to eat lunch: that is, if the second time is at least 45 minutes after the first time. Otherwise the method should return false.

You may assume that all parameter values are valid: the hours are both between 0 and 23, and the minute parameters are between 0 and 59. You may also assume that both times represent times in the same day, e.g. the first time won't represent a time today while the second time represents a time tomorrow. Note that the second time might be earlier than the first time; in such a case, your method should return false.

Here are some example calls to your method and their expected return results:

Call	Value Returned
<code>enoughTimeForLunch(11, 00, 11, 59)</code>	true
<code>enoughTimeForLunch(12, 30, 13, 00)</code>	false
<code>enoughTimeForLunch(12, 30, 13, 15)</code>	true
<code>enoughTimeForLunch(14, 20, 17, 02)</code>	true
<code>enoughTimeForLunch(12, 30, 9, 30)</code>	false
<code>enoughTimeForLunch(12, 00, 11, 55)</code>	false

```
public static boolean enoughTimeForLunch (int h1, int m1, int h2, int m2){
    int time1 = h1*60+m1;
    int time2 = h2*60+m2;
    return(time2-time1>=45);
}
```

b. (10 points) Write a static method named `DigitReverse` that receives an integer value passed by the caller and returns the reversed format of that number in an integer. You may assume that the number is non-negative. For example: A nonnegative number: 29107 will return 70192

```
public static int DigitReverse(int x){
    int r = 0;
    while(x!=0){
        int d = x%10;
        r*=10;
        r+=d;
        x=x/10;
    }
    return r;
}
```

c. (10 points) Write a static method that takes an integer `badLength` and a String array `words` as arguments. Your method should find all of the Strings in the array that have a length equal to `badLength`, delete it, and then return it.

For example, if `words` is {"harry", "potter", "is", "a", "wizard"} and `badLength` is 6, then the method should return the record to be {"harry", "is", "a"}.

```

public static String [] CutArray (int badLength, String [] words){
int size = 0;
for (int i=0; i<words.length; i++)
if(words[i].length()!=badLength)size++;

String [] ret = new String[size];
int k=0;
for (int i=0; i<words.length; i++)
if(words[i].length()!=badLength)
    ret[k++]=words[i];
return ret;
}

```

### 3) Complete Problem (17 points, 20 minutes)

a. Write a class Time that presents a time of day. It has the **private** attributes for the hour and minute, in the 24-hour notation formation. That is, the hour value ranges from 0 to 23, where the range 0 to 11 represents a time before noon. The minute value ranges from 0 to 59. Write the necessary constructor, mutator, and accessor to support the following program in main(). Note that the print-out of Time object will adopt the 12-hour notation formation, i.e., hh:mm AM/PM. See some samples in the following demonstration.

```

----jGRASP exec: java Time
Input your arrivale time (hhmm):
6 0
6:0 AM
Now your arrival time is set as 7:30 AM
----jGRASP: operation complete.

----jGRASP exec: java Time
Input your arrivale time (hhmm):
9 45
9:45 AM
9:45 AM is between 7:30 AM and 10:30 AM
----jGRASP: operation complete.

----jGRASP exec: java Time
Input your arrivale time (hhmm):
14 35
2:35 PM
Now your arrival time is set as 10:30 AM
----jGRASP: operation complete.
L

```

```

import java.util.Scanner;

public class Time{
private int hour, minute;

public Time (int x, int y){
hour = x;
minute = y;
}

public boolean isBetween(Time x, Time y){
if(isEarlierThan(x)) return false;
else if( y.isEarlierThan(this)) return false;
else return true;
}

public int getHour(){
return hour;
}
}

```

```

public int getMinute(){
return minute;
}

public void reSet(Time x){
hour=x.getHour();
minute=x.getMinute();
}

public String toString(){
String ret;
if(hour>11){
ret = ""+(hour-12);
ret += ":"+minute+" PM";
}else {
ret = ""+hour;
ret += ":"+minute+" AM";
}
return ret;
}

public boolean isEarlierThan(Time x){
if(hour < x.getHour()) return true;
else if (hour==x.getHour()&&minute < x.getMinute()) return true;
else return false;
}

public static void main(String [] args){
int hour, minute;
Time early = new Time(7,30);
Time late = new Time(10,30);
Scanner kb = new Scanner(System.in);
do{
System.out.println("Input your arrivale time (hhmm): ");
hour = kb.nextInt();
minute = kb.nextInt();
}while (!(0<=hour && hour <=23)||!(0<=minute && minute <=59));
Time your_arrival = new Time(hour, minute);
System.out.println(your_arrival);

if(your_arrival.isBetween(early, late)){
System.out.println(your_arrival+" is between "+early+" and "+late);
}else if (your_arrival.isEarlierThan(early)){
your_arrival.reSet(early);
System.out.println("Now your arrival time is set as "+your_arrival);
}else {
your_arrival.reSet(late);
System.out.println("Now your arrival time is set as "+your_arrival);
}
}
}
}

```