

CPSC 233 Midterm Review: Winter 2013

Written questions:

Question 1

Refer to the code in class 'MyFormatter' and class 'Driver'

```
public class MyFormatter {
    private String first;
    private String last;
    private Scanner in;
    public MyFormatter () {
        first = null;
        last = null;
        in = new Scanner(System.in);
    }

    public void promptNames () {
        System.out.print("Enter first name: ");
        first = in.nextLine();
        System.out.print("Enter last name: ");
        last = in.nextLine();
    }

    public String formatName () {
        String name;
        // Write your answer to Part (I) here

        return name;
    }
}

public class Driver {
    public static void main (String [] args) {
        MyFormatter aFormatter = new MyFormatter();
        String formatted;
        // Write your answer to Part (II) here

        System.out.println(formatted);
    }
}
```

Part (I):

In the body of method 'formatName':

- write the call to method 'promptNames'
- write the code so that the variable 'name' will combine the 'first' and 'last' attributes in the following manner: <last> <tab> <first>
e.g., first= "charlie", last= "sheen", name= "sheen charlie"

Part (II):

In the body method 'main': Write the call to method 'formatName' and have the return value stored in the variable 'formatted'.

Question 2

Write an accessor/get and mutator/set method for the following class definition.

```
public class Student
{
    private int identificationNumber;

    public Student ()
    {
        identificationNumber = -1;
    }

    // Accessor

    // Mutator

}
```

Question 3

For this question refer to the following program.

```
public class Driver {
    public static void main (String [] args) {
        Manager aManager = new Manager();
        aManager.add(45000);
        aManager.add(56000);
        aManager.add(45000);
        aManager.add(24500);
        aManager.add(75000);
        aManager.display();
        aManager.remove(100000);
        aManager.display();
        aManager.remove(75000);
        aManager.display();
        aManager.remove(45000);
        aManager.display();
    }
}
```

```
public class Manager {
    public final static int MAX = 10;
    public static final int EMPTY = -1;
    private int [] salaries;
    private int lastElement;
    public Manager () {
        int i;
        lastElement = EMPTY;
        salaries = new int [MAX];
        for (i = 0; i < MAX; i++)
            salaries[i] = EMPTY;
    }

    public boolean isEmpty () {
        if (lastElement <= EMPTY)
            return true;
        else
            return false;
    }
    public boolean isFull ()
    {
        if ((lastElement+1) == MAX)
            return true;
        else
            return false;
    }

    public void display () {
        int i;
        for (i = 0; i <= lastElement; i++)
            System.out.println(salaries[i]);
        System.out.println("");
    }

    // Adds new elements to the end of the list.
    public void add (int aSalary) {
        if (isFull() == false)
            salaries[++lastElement] = aSalary;
        else
            System.out.println("Can't add: list is already full");
    }

    // Returns index of first matching salary or EMPTY if no matches.
    private int indexAt (int aSalary) {
        // Assume that this code works
        return index;
    }
}
```

```
public void remove (int aSalary) {  
    << Write your answer here >>  
}
```

<< End of answer space >>

```
}
```

Write the remaining code for the 'remove' method that, given the index of the first match from the the 'indexOf' method it will remove the element at this index and retain the current ordering in the list.

Question 4

For this question you are refer to code in class 'Car' and class 'Start'. Modify the code in the 'main' method so that an array of 'SIZE' elements is created with each element referring to a Car object. Set the price of each car as follows: <First element = 10000> <Second element = 20000> <Third element = 30000>....<Tenth element = 100000>. Finally the program should display the attributes of each car object, one car per line with the model information displayed first and the price information displayed second.

```
public class Start {
    public static void main(String [] args) {
        final int SIZE = 10;
        int i;
        Car myCollection [];
        << Write your answer here >>
    }
}
```

<< End answer space >>

```
}
}
```

```
public class Car {
    private String model;
    private int price;

    public Car () {
        setModel("No name");
        setPrice(-1);
    }

    public String getModel () {
        return model;
    }

    public int getPrice () {
        return price;
    }

    public void setModel (String aModel) {
        model = aModel;
    }

    public void setPrice (int aPrice) {
        price = aPrice;
    }
}
```

Question 5

Write the output of the following program in the space provided.

```
public class Tracer {
    private int x;
    private int y;

    public Tracer() {
        x = 7;
        y = 13;
    }

    public Tracer(int x, int y) {
        this();
        this.x = x;
        y = y * 2;
    }

    public void display() {
        System.out.println(x + " " + y);
    }

    public void method() {
        int x = 2;
        x = x * 10;
        y = x + y;
    }

    public void method(int a) {
        x = a;
        y = this.y * 2;
    }
}

public class Driver {
    public static void main(String [] args) {
        Tracer aTracer = new Tracer();
        aTracer.display();
        aTracer = new Tracer(888,666);
        aTracer.display();
        aTracer.method();
        aTracer.display();
        aTracer.method(707);
        aTracer.display();
    }
}
```

<< Write your answer here >>

Multiple choice questions:

1. What will be the output of the following program?

```
public class Trace1 {  
    public static void main (String [] args) {  
        int num1 = 2;  
        int num2 = 4;  
        int num3 = ++num1 * num2++;  
        System.out.println(num3);  
    }  
}
```

- a) 0
 - b) 8
 - c) 12
 - d) 15
 - e) None of the above
2. Which of the following is an instance of class Foo?

```
public class Driver {  
    public static void main (String [] args) {  
        Foo [] foey;  
        foey = new Foo [14];  
    }  
}
```

```
public class Foo {  
    private int num;  
    public Foo () { num = 0; }  
    public int getNum () { return num; }  
    public void setNum (int n) { num = n; }  
}
```

- a. num
- b. foey
- c. foey[0]
- d. foey[13]
- e. None of the above is an instance of class Foo.