CPSC 233 Midterm review

Short answer 1:

What is the output of the following program?

```
public class Series {
    public static void main(String [] args) {
        int i = 1;
        while (i <= 100) {
            if (i < 10) {
                System.out.print(i + " ");
                i = i + 1;
            }
            else if (i < 25) {
                System.out.print(i + " ");
                i = i * 2;
            }
            else if (i <= 75) {
                System.out.print(i + " ");
                i = i + 20;
            }
            else {
                System.out.print(i + " ");
                i = i + 2;
            }
        }
    }
}
```

<< **Write your answer here** >>

Short answer 2:

What is the output of the following program?

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

    public Tracer() {
        x = 8;
        y = 88;
    }

    public void setX(int num) { x = num; }
    public void setY(int num) { y = num; }
    public void m1() {
        int t = x;
        x = y;
        y = t;
    }

    public void m2() {
        int x = y;
        y = x;
    }
    public void show() {  System.out.println(x + " " + y);  }
    public void showIt() {  System.out.println(x + y);  }
}
public class TracerDriver {
    public static void main(String [] args) {
        Tracer t = new Tracer();
        t.show();
        t.setX(13);
        t.setY(7);
        t.show();
        t.m1();
        t.show();
        t.m2();
        t.showIt();
    }
}
```

<< **Answer** >>

Short answer 3:

Part (a)

Draw out a UML class diagram for class `Person` (full version of class diagram with all details).

```
public class Person {
    private int age;
    public int getAge() {  return(age);  }
    public void setAge(int anAge) {  age = anAge;  }
}
```

Part (b)

Draw a UML class diagram showing the relationship between children and their parents. You are to use a single class diagram. You don't however have to worry about attributes, just focus on representing the relationship. You can assume that each person has up to two parents and that parents can have zero up to 50 children.

Short answer 4:

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


   }
}
```

Short answer 5:

In the space provided you are to specify the output of the following program.

```
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** >>

Short answer 6:

Using the code for class '`Driver`' and class '`Survey`' you are to implement the following capabilities. The solution will be implemented in the methods of the survey class. It will start by asking 3 questions. At the start of the simulation, the person has a 0% chance of catching the illness. The answer to each question will determine the additional probability of catching the illness.

Age:

- 21 years or younger: 0% added to probability
- 22 to 45 years old: 2% added to probability
- 46 - 65 years old: 4% added to probability
- Over 65 years old: 8% added to probability

Drinking frequency:

- Don't drink: 0% added to probability
- One drink a month: 2% added to probability
- One drink a week: 4% added to probability
- One or more drinks a day: 8% added to probability

Smoking frequency:

- Non-smoker: 0% added to probability
- One pack a week: 2% added to probability
- One pack day: 4% added to probability
- More than one pack a day: 8% added to probability

After asking these three questions the program will then determine if the person has caught the illness or not (display result to user).

```
public class Survey {
    public static final int NO_EFFECT = 0;
    public static final int LOW_EFFECT = 2;
    public static final int MEDIUM_EFFECT = 4;
    public static final int HIGH_EFFECT = 8;
    public static final int NUMBER_PERCENTAGES = 100;
    private Scanner in;
    private Random aGenerator;




    public Survey() {
        in = new Scanner(System.in);
        aGenerator = new Random();
        << Additional constructor code here >>




    }

    << Start of answer space >>
    public void askDrinking() {
        System.out.println("(0) Don't drink");
        System.out.println("(1) One drink a month");
        System.out.println("(2) One drink a week");
        System.out.println("(3) One or more drinks a day");


    }

    public void askSmoking() {
        System.out.println("(0) Non smoker");
        System.out.println("(1) One pack a week");
        System.out.println("(2) One pack day");
        System.out.println("(3) More than one pack a day");


    }
```

```
    public void askAge() {
        System.out.println("(0) 21 years or younger");
        System.out.println("(1) 22 to 45 years old");
        System.out.println("(2) 46 - 65 years old");
        System.out.println("(3) Over 65 years old");


    }
```

<< **End of answer space** >>

```
}


public class Driver {
    public static void main(String [] args) {
        Survey aSurvey = new Survey();
        aSurvey.start();
    }
}
```

Multiple choice questions:

For Questions 1 – 2 you are to specify how many times the corresponding loop will execute.

1)

```
int i = 10;
while (i <= 5) {
    System.out.print(i);
    i = i + 1;
}
```

a. 0
b. 1
c. 6
d. 10
e. This is an endless loop

2)

```
for(int i=100; i<0; ++i) {
    System.out.print(i);
    i = i + 1;
}
```

a. 0
b. 1
c. 100
d. 101
e. This is an endless loop

For Questions 3 – 4 you are to specify the output of the corresponding program.

3)

```
char a = '-';
switch (a) {
    case '1':
        System.out.print('a');
        break;
    case '2':
        System.out.print('b');
        break;
    case '3':
        System.out.print('c');
        break;
    case '4':
        System.out.print('d');
        break;
 }
 System.out.print("e");
```

a. a
b. ae
c. 1
d. 1e
e. e

4)

```
int b;
Random aGenerator = new Random();
b = aGenerator.nextInt(100)+1;
if ((b <= -1) || (b > -1))
    System.out.println("Doh!");
else
    System.out.println("Woohoo!");
```

a. a
b. Doh
c. Doh!
d. Doh!
   Woohoo!
e. None of the above

5) Which of the following is a Java constant?
a. int x = 1;
b. long x = 3;
c. final char x = ‘x’;
d. constant int x = 1’
e. CONSTANT = 12;

6) Which of the following is the correct minimum number of statements needed to *declare and initialize an array* with guaranteed starting values?

```
a. anArray = [];

b. anArray = [1,2,3];

c. int [] anArray;

d. int [] anArray = new int[3];

e. int [] anArray = new int[3];
   for (int i = 0; i < 3; i++) { anArray[i] = -1; }
```

For Questions 7 & 8 please refer to the following program.

```
public class Calls {
    public void meth1 () {
        System.out.println("meth1");
    }

    public void meth2 () {

    }
}

public class Driver {
    public static void main (String [] args) {
        Calls aCall = new Calls();

    }
}
```

7) Based on what you see in the class definition: Within the body of '`meth2()`' which of the following is the proper way to invoke '`meth1()`'?
   a. meth1();
   b. this.meth1();
   c. aCall.meth1();
   d. (a) & (b)
   e. All of the above

8) Based on what you see in the class definition: Within the body of '`main()`' which of the following is the proper way to invoke '`meth1()`'?
   a. meth1();
   b. this.meth1();
   c. aCall.meth1();
   d. (a) & (b)
   e. All of the above

Ideas for practice trace questions:

Take the existing programs and modify them e.g., (assuming there exists an attribute 'a') make '`a = 2`' become '`this.a = 2`'. As long as the result is syntactically valid then you can use the modified program for extra practice. Just make sure that you try your hand at a hand trace before running the program otherwise you will get nothing out of the exercise.

Other sources of ideas

Finally you can look through the recommended paper text book as well other Java textbooks you find on the library 'Safari' website: http://proquest.safaribooksonline.com.ezproxy.lib.ucalgary.ca/

(In the latter case just make sure that you select generic Java textbooks, ones that list topics which we have talked about such as "NetBeans" or "Enterprise" in the title will likely be too specific).