Midterm Examination 2
CS170: Introduction to Computer Science

Observe the Emory College Honor Code while taking this test.

Question 1. (20 pts)

Give brief answers to the questions.

1. What is the terminology for a method that do not have an implicit parameter (I’m looking for a technical term that describes such method) ?
   
   Class method

2. What is the terminology for a method that has an implicit parameter (I’m looking for a technical term that describes such method) ?
   
   Instance method

3. What keyword is used to specify a constant in Java ?
   
   final

4. Show a different way to write:

   \[
   \text{String greeting} = "Hello";
   \]

   \[
   \text{String greeting} = \text{new String("Hello")};
   \]

5. What possible values can you store in a boolean variable ?
   
   true, false

6. How many statements are in the following program fragment:

   \[
   \text{help} = \text{x}; \quad \text{3 statements}
   \]

   \[
   \text{x} = \text{y};
   \]

   \[
   \text{y} = \text{help};
   \]

7. How many statements are in the following program fragment:

   \[
   \text{if (x > y)} \quad \text{3 statements}
   \]

   \[
   \text{help} = \text{x};
   \]

   \[
   \text{x} = \text{y};
   \]

   \[
   \text{y} = \text{help};
   \]
8. How many statements are in the following program fragment:

```java
if ( x > y )
    1 statement
{    help = x;
    x = y;
    y = help;
}
```

9. What is the name of the method that is used to compare if two strings are identical in content?

`equals`

10. What operator do you use to test if two object variables reference to the same object?

`==`
Question 2. (20 pts)

Suppose you are given the following class definition:

```java
public class Q2
{
    public Q2()
    {
        ... the body is not relevant to the question as is omitted...
    }

    public static void method1()
    {
        ... the body is not relevant to the question as is omitted...
    }

    public static void method2(int x)
    {
        ...
    }

    public static void method3(Q2 x)
    {
        ...
    }

    public void method4()
    {
        ...
    }

    public void method5(int x)
    {
        ...
    }

    public void method6(Q2 x)
    {
        ...
    }
}
```

In addition, the following variables are defined:

```java
Q2 help1 = new Q2();
Q2 help2 = help1;
Q2 help3;
```
Questions

1. Will Java report any error for the following statement? (2 pts)

   Q2.method1();         Answer: No
   If yes, what's wrong:________________________

2. Will Java report any error for the following statement? (2 pts)

   Q2.method3();         Answer: Yes
   If yes, what's wrong: Needs parameter

3. Will Java report any error for the following statement? (2 pts)

   Q2.method4();         Answer: Yes
   If yes, what's wrong: Needs object

4. Will Java report any error for the following statement? (2 pts)

   help1.method4();      Answer: No
   If yes, what's wrong:________________________

5. Will Java report any error for the following statement? (2 pts)

   help1.method6(help1); Answer: No
   If yes, what's wrong:________________________

6. Will Java report any error for the following statement? (2 pts)

   help2.method6(help1); Answer: No
   If yes, what's wrong:________________________

7. Will Java report any error for the following statement? (2 pts)

   help3.method6(help1); Answer: Yes
   If yes, what's wrong: help3 not initialized
8. Give an example of a correct invocation of method `method2()`: (3 pts)
   `Q2.method2(4);`

9. Give an example of a correct invocation of method `method5()`: (3 pts)
   `help1.method5(4);`
Question 3 (20 pts)

The following programs are to test if you understand Java syntax. I have deliberately written the programs without using any indentation. You need to determine how the programs are "read" and if they are correct.

- Write down the output of this program in the space provide if the program does not contain any errors. Write "Error" if the program contains an error.

```java
public class Q3a {
    public static void main(String[] args) {
        int x = 10;
        if (x > 50 || x < 20) System.out.print("A");
        else System.out.print("B");
    }
}
Output: A
```

- Write down the output of this program in the space provide if the program does not contain any errors. Write "Error" if the program contains an error.

```java
public class Q3b {
    public static void main(String[] args) {
        int x = 10;
        if (x > 50 && x < 20) System.out.print("A");
        System.out.print("B");
    }
}
Output: B
```
• Write down the output of this program in the space provide if the program does not contain any errors. Write "Error" if the program contains an error.

```java
public class Q3c {
    public static void main(String[] args) {
        int x = 10;

        if (x > 50 || x < 20) {
            System.out.print("A");
        } else {
            System.out.print("B");
        }
    }
}

Output: error - else with no if
```

• Write down the output of this program in the space provide (this program does not contain any error).

```java
public class Q3d {
    public static void main(String[] args) {
        int x = 45;

        if (x > 50 || x < 20) {
            if (x > 20 && x < 50) {
                System.out.print("A");
            } else {
                System.out.print("B");
            }
        } else {
            if (x < 20 || x > 40) {
                System.out.print("C");
            } else {
                System.out.print("D");
            }
        }
    }
}

Output: C
```
• Write down the output of this program in the space provide (this program does not contain any error).

    public class Q3e
    {
        public static void main(String[] args)
        {
            int x = 70;

            if ( x > 50 || x < 20 )
                System.out.print("A");
            if ( x > 0 && x < 100 )
                System.out.print("B");
            if ( x < 20 || x > 40 )
                System.out.print("C");
            else
                System.out.print("D");
        }
    }

    Output: A B C
Question 4 (20 pts)

The following programs are to test if you understand Java syntax. I have deliberately written the programs without using any indentation. You need to determine how the programs are "read".

- Write down the output of this program in the space provide. If the loop does not end, write as answer: “infinite loop”

```java
public class Q4a {
    public static void main(String[] args) {
        int x = 10;
        while ( x < 20 || x > 100 )
            x = 100 - x;
        x = 2*x;
        System.out.print(x);
    }
}
Output: 180
```

- Write down the output of this program in the space provide. If the loop does not end, write as answer: “infinite loop”

```java
public class Q4b {
    public static void main(String[] args) {
        int x = 10;
        while ( x < 20 || x > 100 )
            x = 100 - x;
        x = 2*x;
        System.out.print(x);
    }
}
Output: infinite loop
```
• Write down the output of this program in the space provided. If the loop does not end, write as answer: “infinite loop”

    public class Q4c
    {
    public static void main(String[] args)
    {
        int x = 0;
        int i;
        for (i = 0; i < 10; i++)
            x = x + 1;
        x = 2*x;
        System.out.print(x);
    }
    }

    Output: 20

• Write down the output of this program in the space provided. If the loop does not end, write as answer: “infinite loop”

    public class Q4d
    {
    public static void main(String[] args)
    {
        int x = 0;
        int i;
        for (i = 0; i < 10; i++)
            x = x + 1;
        x = 2*x;
        System.out.print(x);
    }
    }

    Output: 2
Write down the output of this program in the space provided. If the loop does not end, write as answer: “infinite loop”

```java
public class Q4e {
    public static void main(String[] args) {
        int x = 0;
        int i;

        for (i = 0; (x < 20 || x > 100); i++)
            x = x + i;
        x = 2*x;

        System.out.print(x);
    }
}
```

Output: 42
Question 5 (20 pts)

In this question, we will design and implement a "LightBulb" object. A LightBulb has a lifetime and an age. While the age is less than or equal to its lifetime, the LightBulb is operational, and otherwise it is not operational. While a LightBulb object is operational, it can be toggled (switched) on and off:

```
age = 0
lifetime = 576
```

```
toggle()
```

```
operational
off
```

```
age = 1
lifetime = 576
```

```
toggle()
```

```
operational
on
```

```
age = 2
lifetime = 576
```

```
toggle()
```

```
operational
off
```

Each time the LightBulb is turned on, its age is increased by 1. The age is not increased if the LightBulb is turned off (see figure above: the second `toggle()` does not increase the age).

When the age of a LightBulb is greater than its lifetime, the LightBulb becomes not operational and it will remain "off":

```
age = 575
lifetime = 576
```

```
toggle()
```

```
operational
off
```

```
age = 576
lifetime = 576
```

```
toggle()
```

```
operational
on
```

```
age = 576
lifetime = 576
```

```
toggle()
```

```
operational
off
```

```
age = 577
lifetime = 576
```

```
toggle()
```

```
NOT operational
off
```

The second figure depicts the fact that the LightBulb has a lifetime of 576 and it has been turned on (and off) 575 times. When the LightBulb is turned on one more time, its age is equal to the lifetime and it will be on. When its toggled off and on again, the LightBulb will not be “on” any more because the age exceeds its lifetime.

The LightBulb object has 4 method:

- **LightBulb(int n):** constructs a light bulb with a lifetime that is *random integer number* drawn from 1 and n. The age of a new LightBulb object is 0 and the LightBulb object is “off” when it is first constructed.

- **getRemainingLife():** returns the number of times that the LightBulb can be turned
on until it becomes not operational. The method must returns 0 if the age of the LightBulb is equal to or exceeds its lifetime.

- **toggle()**: toggles the light bulb. If the age of the LightBulb exceeds the lifetime, the toggle() method has no effect on the LightBulb (the LightBulb remains “off”). If the age is less than or equal to the lifetime, then when the LightBulb is currently “on”, then toggle() will change it to “off”, and vice versa.

- **isOn()**: returns true if the LightBulb is “on”. Remember that the LightBulb can be “on” or “off” when it is operational (the state is changed by the toggle() method) and the LightBulb is “off” when its age exceeds its lifetime.

To give you maximum freedom in the design of the LightBulb class, I will let you define your own instance variables.

**Questions:**

- Write only the constructor method LightBulb(int n) for the LightBulb class in the following class definition for LightBulb. The constructor is used to construct a LightBulb that has a lifetime that is a random integer value from 1 to n, with a starting age of 0 and in the “off” state.

  You must also define all the necessary instance variables as part of the answer. (4 pts)

  Hint: my solution keeps information on the age, the lifetime and the state (on or off) of the light bulb.

```java
import java.util.Random;

public class LightBulb
{
  public LightBulb(int n)
  {
    Random gen = new Random();

    lifetime = gen.nextInt(n) + 1;
    age = 0;
    on = false;
  }

  private int age;
  private int lifetime;
  private boolean on;
}
```
• Write only the method `getRemainingLife()` for the `LightBulb` class in the following class definition for `LightBulb`. When the method `getRemainingLife()` is called, it must return the number of times that the `LightBulb` can be turned on before it becomes not operational.

NOTE: the `getRemainingLife()` method must return 0 when the age is greater than or equal to the life time of the light bulb. (4 pts)

NOTE: you don’t need to define the instance variables inside this method. I will assume that the instance variables in your answer in the previous question are defined.

```java
public class LightBulb {
    public int getRemainingLife() {
        return lifetime - age;
    }
}
```

• Write only the method `isOn()` for the `LightBulb` class in the following class definition for `LightBulb`. When the method `isOn()` is called, it returns true if the `LightBulb` object is “on” and otherwise, it returns false: (4 pts)

```java
public class LightBulb {
    public boolean isOn() {
        return on;
    }
}
```

• Write only the method `toggle()` for the `LightBulb` class in the following class definition for `LightBulb`. See the description of the problem in how `toggle()` works. (It’s too long to repeat). (4 pts)

```java
public class LightBulb {
    public void toggle() {
        if ( age <= lifetime ) {
            on = !on;
            if ( on )
                age++;
        }
    }
}
```
else
    on = false;
}
}

- Write a program that perform the following operations (in the exact sequence as given):

  1. Construct a LightBulb object named deskLight whose lifetime is a random number between 1 and 5000.
  2. Toggle the deskLight on and off for 2000 times (so you toggle a total of 4000 times, and the deskLight is “off” when you're done)
  3. Toggle the deskLight one more time.
  4. Print the remaining life of the deskLight when it is “on” and otherwise, print the message "Need a new lamp".

    ```java
    public class LightBulbTester {
        public static void main(String[] args) {
            int i;
            LightBulb deskLight = new LightBulb(5000);

            for (i = 0; i < 2000; i++)
            {
                deskLight.toggle();
                deskLight.toggle();
            }

            deskLight.toggle();

            if (deskLight.isOn())
                System.out.println(deskLight.getRemainingLife());
            else
                System.out.println("Need new lamp");
        }
    }
    ```