Midterm Exam

Course CS2040, Spring 2020, American University of Paris

Student Name and ID number: ____________________

The use of book, notes or internet access is not allowed.

1 Part I (2 points each)

1. What is wrong in the following code?

```java
class One {

   public static void main(String[] args) {
   MyClass myObject = new MyClass();
   myObject.print();
   }

}


class MyClass {
   String s;

   MyClass (String s) {
      this.s = s;
   }

   public void print() {
      System.out.print(s);
   }
}
```

Answer:

____

____

____

____________________ □

2. What is the printout of the following code? Why?

```java
public class Two {
   private boolean x;
   private String a;

   public static void main(String[] args) {
      Two myObject = new Two();
      System.out.println(myObject.x);
      System.out.println(myObject.a);
   }
}
```

Answer:

____

____

____

____ □

3. Show the output of the following program:

```
public class Three {

   public static void main(String[] args) {
      Count myCount = new Count();
      int times = 0;
      for (int i = 0; i < 20; i++){
         increment(myCount, times);
      }
      System.out.println("count is " + myCount.getCount());
      System.out.println("times is " + times);
   }

   public static void increment(Count c, int times) {
      c.inc();
      times++;
   }
}
```

```
public class Count {
   private int count;

   Count(int c) {
      count = c;
   }

   Count() {
      count = 1;
   }

   public void inc() {
      count++;
   }

   public int getCount() {
      return count;
   }
}
```

Answer:

4. Show the output of following program and explain it.

```
public class Four {
   public static voi d main(String[] args) {
   ClassA a = new ClassA(3);
   }
}

class ClassA extends ClassB {
   public ClassA (int t) {
      System.out.println ( "ClassA’s constructor is invoked");
   }
}

class ClassB {
   public ClassB() {
      System.out.println ( "ClassB’s constructor is invoked");
   }
}
```

Answer:

5. In Java, the equals() method of the class String overrides the equals() method of the class Object and it compares the two given strings: if the contents of both strings are same then it returns true, if not all characters are matched then it returns false.

   Show the output of the following code by explaining it.

```java
public class Test {
    public static void main(String[] args) {

        String s1 = new String("a string");
        String s2 = new String("a string");
        String s3 = new String("another string");
        String s4 = s1;

        System.out.println(s1 == s2);
        System.out.println(s1 == s3);
        System.out.println(s1 == s4);
        System.out.println(s2 == s4);

        System.out.println(s1.equals(s2));
        System.out.println(s1.equals(s3));
        System.out.println(s1.equals(s4));
        System.out.println(s2.equals(s4));
    }
}
```

Answer:

☐

2 Part II: multiple answer questions (1 point each)

6. What is invoked in order to create an object?
   - ◯ The main method.
   - ◯ A method with a return type.
   - ◯ A method with the void return type.
   - ◯ A constructor.

7. Given the declaration Circle x = new Circle(), which of the following statement is most accurate.
   - ◯ x contains an object of the Circle type.
   - ◯ You can assign an int value to x.
   - ◯ x contains a reference to a Circle object.
   - ◯ x contains an int value.

8. Analyze the following code.

```java
public class Test {
   int x;

   public Test(String t) {
      System.out.println("Test");
   }

   public static void main(String[] args) {
      Test test = null;
      System.out.println(test.x);
   }
}
```

   - ◯ The program has a compile error because x has not been initialized.
   - ◯ The program has a compile error because *test* is not initialized.
   - ◯ The program has a runtime *NullPointerException* because *test* is *null* while executing *test.x*.
   - ◯ The program has a compile error because you cannot create an object from the class that defines the object.
   - ◯ The program has a compile error because Test does not have a default constructor.

9. When invoking a method with an object argument, what is passed?
   - ◯ a copy of the object
   - ◯ the object is copied, then the reference of the copied object
   - ◯ the reference of the object
   - ◯ the contents of the object

10. A final Class . . .
   - ◯ cannot have instances.
   - ◯ can be extended.
   - ◯ has methods which cannot be overloaded.
   - ◯ has methods which can be overridden.
   - ◯ none of the above

11. Class static variables . . .
    - ◯ cannot be modified.
    - ◯ cannot be accessed without an object instance.
    - ◯ are shared among all the instances of class.
    - ◯ can be modified only by methods of the same class.

12. Consider the following class

```java
class myClass {
   private final int i=0;
   myClass(){}

   public int getI(){
      return i;
   }
}
```

    Why is the setter for i not defined?
    - ◯ because i is not initialized.
    - ◯ because i is private.
    - ◯ because i is final.
    - ◯ because $i = 0$.

13. A property of a class MyClass with the protected modifier
    - ◯ can be accessed only by the same class.
    - ◯ can be accessed only by all the classes in the same package.
    - ◯ can be accessed only by all the MyClass subclasses.
    - ◯ can be accessed only by all the classes in the same package or MyClass subclasses.

14. Analyze the following code: class Circle private double radius; public Circle(double radius) radius = radius;
    - ◯ The program has a compile error because you cannot assign radius to radius.
    - ◯ The program will compile, but you cannot create an object of Circle with a specified radius. The object will always have radius 0.
    - ◯ The program does not compile because Circle does not have a default constructor.
    - ◯ The program has a compilation error because it does not have a main method.

15. Given the declaration Circle[] x = new Circle[10], which of the following statement is most accurate.
    - ◯ x contains an array of ten int values.
    - ◯ x contains a reference to an array and each element in the array can hold a Circle object.
    - ◯ x contains an array of ten objects of the Circle type.
    - ◯ x contains a reference to an array and each element in the array can hold a reference to a Circle object.

16. Consider the method

```java
int myMethod (boolean b) {some code}
```

    Which of the following methods overload it?

- ◯ int yourMethod (boolean b) {same code}
- ◯ int myMethod (boolean b) {some other code}
- ◯ int MyMethod (boolean b) {same Code}
- ◯ double myMethod (int x) {some other code}

17. Let A_1 and A_2 be classes and I_1 and I_2 interfaces. A class MyClass ...
    - ◯ can extend A_1 and A_2 at the same time.
    - ◯ cannot instantiate I_2 and I_2 at the same time.
    - ◯ can extend A_1 and instantiate I_2 at the same time.
    - ◯ None of the above.

18. A subclass inherits from its superclass the ...
    - ◯ private method
    - ◯ protected method
    - ◯ public method
    - ◯ private and public methods
    - ◯ protected and public methods

19. How is it possible to create an instance an object of an abstract class?
    - ◯ By invoking its constructor.
    - ◯ It is not possible to create an instance of an abstract class.
    - ◯ By invoking a constructor of one of its subclass.
    - ◯ None of the above.

20. Consider the method

```
protected double xMethod (int x) {some code};
```

Which of the following methods override it?
- ◯ private double xMethod (int x) {some other code}
- ◯ protected int xMethod (double x) {some other code}
- ◯ public double xMethod (double x) {some other code}
- ◯ public double xMethod (int x) {some other code}

21. Which of the following is false?
    - ◯ a class with an abstract method has to be abstract.
    - ◯ a class with an abstract method has all its methods abstract.
    - ◯ an interface has only abstract methods.
    - ◯ a subclass can be abstract even if its superclass is concrete.

22. Analyze the following code:

```
public class Test {
   private int t;

   public static void main(String[] args) {
      int x;
      System.out.println(t);
   }
}
```

- ◯ The program compiles and runs fine.
- ◯ The variable t is private and therefore cannot be accessed in the main method.
- ◯ The variable x is not initialized and therefore causes errors.
- ◯ t is non-static and it cannot be referenced in a static context in the main method.
- ◯ The variable t is not initialized and therefore causes errors.