21/05093 – MUMO DENNIS

Task 3 Part 2

1. With relevant examples, explain the following concepts as used in Java programming.

a. Mutable classes.

Explain what is meant by mutable class

A mutable class is a class that can change its internal state after it is created.

b. Immutable classes.

Explain what is meant by immutable class

An immutable class is a class that can not change its internal state

after it is created.

c. Explain the situations where mutable classes are more preferable than immutable classes when writing a Java program.

- ➔ The internal state of your program will be consistent even if you have exceptions.
- ➔ References to immutable objects can be cached as they are not going to change.
- ➔ Immutable classes make it easier to parallelize your program as there are no conflicts among objects.

2.

a) Explain what a String buffer class is as used in Java, the syntax of creating an object of StringBuffer class and Explain the methods in the StringBuffer class.

- ➔ String buffer is a thread-safe, mutable sequence of characters.
- ➔ Syntax for creating an object of StringBuffer class:
  - ◆ `StringBuffer objectName = new StringBuffer(String);`
- ➔ Methods in the StringBuffer class:
  - ◆ length() - returns the length of the string i.e. total number of characters.
  - ◆ reverse() - returns the string in reversed order.
  - ◆ append(argument) - used to append the specified string with another string. It can also be overloaded to support other data types.
  - ◆ capacity() - returns the current capacity.

b) Write the output of the following program.

```
class Myoutput
1. {
2. public static void main(String args[])
3. {
4. String ast = "hello i love java";
5. System.out.println(ast.indexOf('e')+"
"+ast.indexOf('ast')+" "+ast.lastIndexOf('l')+" "+ast
.lastIndexOf('v'));
6. }
7. }
```

Output:

The program does not execute. Hence we have no output due to errors within the code.

c) Explain your answer in (2b) above.

ast.indexOf('ast') caused a semantic error. indexOf() does not take a String argument hence resulting in an error.

d) With explanation, write the output of the

following program. class Myoutput

1. {

2. public static void main(String args[])

3. {

4. StringBuffer bfobj = new

StringBuffer("Jambo"); 5. StringBuffer bfobj1 =

new StringBuffer(" Kenya"); 6.

c.append(bfobj1);

7. System.out.println(bfobj);

8. }

9. }

The program does not execute due to another semantic error.
As from this line 6. `c.append(bfobj1);` There is no variable declaration of “c”.

e) With explanation, write the output of the

following program. class Myoutput

```
1. {
2. public static void main(String args[])
3. {
4. StringBuffer str1 = new StringBuffer("Jambo");
5. StringBuffer str2 = str1.reverse();
6. System.out.println(str2);
7. }
8. }
```

Output:

obmaJ

The output is “obmaj” because the original String “Jambo” has been reversed by the reverse() function.

f) With explanation, write the output of the following program. class Myoutput

```
1. {
2. class output
3. {
4. public static void main(String args[])
5. {
6. char c[]={'A', '1', 'b' ,' ' ,'a' , '0'};
7. for (int i = 0; i < 5; ++i)
8. {
```

```
9. i++;
10. if(Character.isDigit(c[i]))
11. System.out.println(c[i]+" is a digit");
12. if(Character.isWhitespace(c[i]))
13. System.out.println(c[i]+" is a Whitespace character");
14. if(Character.isUpperCase(c[i]))
15. System.out.println(c[i]+" is an Upper case Letter");
16. if(Character.isLowerCase(c[i]))
17. System.out.println(c[i]+" is a lower case Letter");
18. i++;
19. }
20. }
21. }
```

Output:

```
1 is a digit
a is a lower case Letter
```