**CSE 201: Advanced Programming**

**Mid-Semester Exam**

**MM: 41 marks+20 marks bonus**

**Note: For some questions, if certain mentioned keywords are found in the answer we have given marks**

**Question 1.** **(2 marks):** What is the output of the following program:

public class TestDogs {

public static void main(String [] args) {

Dog [][] theDogs = new Dog[3][];

System.out.println(theDogs[2][0].toString());

}

}

class Dog { }

***Ans:*** *NullPointer Exception. theDogs[2][0] has been used without being initialized first.*

***Evaluation: 1 mark for output, 1 mark for reason***

**Question 2. (2 marks):** What is the output of the following program:

class Student {

public void read() {

System.out.println("Reading!");

}

}

class IIIT {

Student student = new Student() {

public void read() {

System.out.println("Reading!!!");

}

public void learn() {

System.out.println("Learning");

}

};

public void learningOutcome() {

student.read();

student.learn();

}

public static void main(String[] args){

IIIT w = new IIIT();

w. learningOutcome ();

}

}

***Ans.*** *Compile time error at student.learn(). Anonymous inner class is extending from Student but the object type has been upcasted to Student. Method learn which is defined in the child class cannot be called from this upcasted object.*

***Evaluation: 1 mark for output, 1 mark for reason***

**Question 3. (2 Marks):** What is the output of the following program

public class Addition {

void PrintSum(int a, int b){

System.out.println(a+b);

}

public static void main(String[] args) {

Addition a = new Addition();

a.PrintSum(10, 20);

}

}

***Ans.*** *30*

***Evaluation: 2 marks for output***

**Question 4. (10 Marks):** Write a Java method that will return the number of bits that will need to be changed in order to convert an integer, X, into another integer, Y and vice versa. The method should accept two different integers as input. For example, if your method is passed the integers 12 and 16 then your method should return a 3

***Ans. One of the many possibly answers:***

*int changed(int a, int b)*

*{*

*int c=a^b;*

*int d= Integer.bitCount(c);*

*System.out.println("Number of bits to be changed=" +d);*

*return d;*

*}*

*Some students could have used toString with radix for binary conversion and then counting the 1’s character-wise.*

**Question 5. (2 Marks):** What is an immutable object? Explain with example.

**Ans.**

*An object is considered immutable if its state cannot change after it is constructed.*

*Example, String once constructed cannot be edited. One can only change reference to another object of string.*

*String a="abc";*

*a.concat("def");*

*System.out.println(a);//prints abc and not abcdef*

*Other examples are Integer, Double etc.*

***Evaluation: 1 mark for ‘cannot be modified after construction’ keywords, 1 mark for example***

**Question 6. (2 Marks):** What is the Difference between JDK and JRE ?

***Ans.*** *JRE - Java Runtime Environment is used to provide runtime environment. It is the implementation of JVM. It contains set of libraries and other files that JVM uses at runtime.*

*JDK-Java Development Kit on the other hand contains JRE along with development tools eg. Javac, java*

*The important keywords that must be covered are JRE contains only the run time environment or the JVM. JDK is superset containing JRE and other development tools.*

***Evaluation: Keywords- compilation and runtime***

**Question 7. (3 Marks):** Differentiate between an Interface and an Abstract class, in terms of inheritance, final and access modifiers.

## *Ans.*

## *A class can only inherit from one abstract class whereas can implement multiple interfaces.*

## *All variables in interface are final. No such restriction for Abstract class*

## *All function/variables are public in interface. No such for Abstract class.*

***Evaluation: 1 mark for each of the above***

**Question 8. (2 Marks):** What is the difference between Iterator and ListIterator ?

**Ans.** *We can traverse in only forward direction using Iterator.Using ListIterator, we can traverse a List in both the directions (forward and Backward).*

*We cannot obtain indexes while using Iterator. We can obtain indexes at any point of time while traversing a list using ListIterator. The methods nextIndex() and previousIndex() are used for this purpose.*

**Only first was covered in class. But we have allowed others. One difference expected.**

**Question 9. (2+2 marks):** When would you use Comparable interface within the context of collections. When do you use Comparator interface?

***Ans.*** *Many sorting functionalities in collections require to compare two objects of a class. Comparable and Comparator are two ways to enable collections class to compare two objects.*

*Comparable is implemented by a class in order to be able to compare object of itself with some other objects. The class itself must implement the interface in order to be able to compare its instance(s). The method required for implementation is compareTo()*

*In some situations, you may not want or allowed to change a class and make it comparable. In such cases, Comparator can be used if you want to compare objects based on certain attributes/fields. The method required to implement is*compare()*.*

***Evaluation: Keywords-Comparator allowed without modifying the original class***

**Question 10. (2 Marks):** In Swing, what is the difference between a Window and a Frame class?

***Ans.*** *A frame is a resizable, movable window with title bar and close button. Usually it contains Panels. It’s derived from a window.*

*A window is a Container that must have a parent Frame mentioned in the constructor.*

***Evaluation: Keywords-Window contains frame***

**Question 11. (2 Marks):** How can a component of type Button in JavaFX handle its own events?

**public** **class** MyButton **extends** Button **implements** EventHandler<ActionEvent>

{

**public** MyButton()

{

setOnAction(**this**);

}

@Override

**public** **void** handle(ActionEvent arg0) {

System.***out***.println("Hello World!!");

}

}

***Evaluation: Iimportant statements such as setOnAction(this) , the class implements EventHandler<> and an appropriate handle method ,if found have been given marks.***

**Question 12. (2 marks):** What happens when an exception is thrown by the main method in a program?

**Ans.** The program exits, printing the stacktrace on the console. For the purpose of exception, we can imagine the main function to be running in an imaginary try catch block like following:

try {

loadClasses();

runInitializers();

main(argv);

System.exit(0);

} catch (Throwable e) {

e.printStackTrace();

System.exit(-1);

}

***Evaluation: Keywords-exits and prints stacktrace***

**Question 13. (2 marks):** What is/are the problem(s) (if any) with the following program:

public class MyProgram {

public static void main(String[] args) {

try{

foo1();

}catch(Exception e){

e.printStackTrace();

}catch(NullPointerException e){

e.printStackTrace();

}

foo2();

}

public static void foo1() { … }

public static void foo2() throws IOException { … }

public static void foo3() throws NullPointerException { … }

public static void foo3() throws Exception { … }

}

**Ans.**  *foo3 has been defined twice which would give error. foo2 in main should be in trycatch block with IOException. Catch(Exception e) already catches NullPointer Exception, so recatching it in other catch block gives error- compile time. If required the two catch must be reordered so that NullPointer thing comes before more general ones.*

***Evaluation: 3 mentioned errors expected. 0.5 deducted per missing error***

**Question 14. (2 Marks)** When I try to use the following program to write to file, resulting file is empty. Explain why?

public class MyFileWriter {

public static void main(String[] args) throws Exception {

FileOutputStream myfos = new FileOutputStream("foo.txt");

PrintWriter mypw = new PrintWriter(myfos);

mypw.print(true);

mypw.println("blabla");

}

}

**Ans.** *The JVM or OS tries to buffer the data to reduce accessing the frequent access of hard disk or other media. If you do not flush and close the stream, the data might be lost. Adding these lines at the end would resolve the issue.:*

*mypw.close();*

***Evaluation: Keywords-flushing or closing the stream***

**Question 15. (2 Marks)** If I need to copy the contents of one file to another, which streams should I use for best performance in terms of time taken to complete the operation.

***Ans.*** *BufferedInputStream/BufferedReader to read content from a file. BufferedOutputStream/BufferedWriter to write content to a file.*

*No Program required…*

**Bonus:**

**Question 16. (10 Marks):** The following program will fail to find the correct answer. Give reasons

boolean FindOdd(int num) {

return num % 2 == 1;

}

***Ans.*** *Would not work for negative integers as negative integers on mod 2 give -1 eg. -7mod2=-1*

A simple fix is to check against != 0 rather than == 1.

***Evaluation: 5 marks for mentioning won’t work for negative integers. Other 5marks for explaining why i.e modulo gives -1.***

**Question 17.(10 Marks):** It can be argued that character arrays are safer to store the sensitive information such as password etc. compared to Strings in a Java program. Can you reason why it should be so?

**Ans.** *Strings are immutable and may not be destroyed until by garbage collector. So the password stored as string may exist in memory making it insecure.*

***Evaluation: Keywords-char mutable, string immutable - stays in memory, garbage collector***