OOP II Sem TA2 Quiz

Total Number of Questions: 30
Duration : 25 minutes
5 minutes buffer

nimjess@rknec.edu Switch account

Draft saved

* Indicates required question

Email *

Record **nimjess@rknec.edu** as the email to be included with my response

Section *

A

Roll No. (Write only roll no., do not write the section) *

77

Name of Student *

Sujal Nimje

Output will be... *

```
import java.io.FileInputStream;
import java.io.ByteArrayInputStream;
      public class Main
      {
            public static void main(String[] args)        {
              String obj  = "hut";
            byte b[] = obj.getBytes();
            ByteArrayInputStream obj1 = new ByteArrayInputStream(b);
            for (int i = 0; i < 2; ++ i)
            {
              int c;
              while((c = obj1.read()) != -1)
              {
                        if(i == 0)
                {
                          System.out.print(Character.toUpperCase((char)c));
                 }
              }
              System.out.print(obj);
            }
          }
      }
```

- (●) HhhUhhThh
- ( ) HUThhh
- ( ) HUThuthut
- ( ) HhUhTh

Select the packages which contains classes and interfaces used for input & output operations of file? *

- [x] java.io
- [ ] java.lang
- [ ] java.net
- [ ] java.util

Which is the closest common parent of RuntimeException, Error, IOException and ClassNotFoundException is: *

- (●) Throwable
- ( ) Object
- ( ) Exception
- ( ) Catchable

*

Which task can be performed by the methods supported by OutputStreams

a) closing streams
b) flushing streams
c) reading bytes
d) writing bytes

- [ ] a, b and c only
- [ ] All a, b, c and d
- [ ] b, c and d only
- [x] a, b and d only

Output will be..... *

```
class Main {
        public static void main(String args[])   {
          try  {
            System.out.print("Welcome to " + " " + 1 / 0);
              }
          catch(ArithmeticException e)      {
        System.out.print("RCOEM");
          }
       }
    }
```

- ○ RCOEM
- ◉ Welcome to RCOEM
- ○ Welcome to
- ○ Welcome toRCOEM

In Java, which keyword is used to manually throw an exception? *

- [x] throw
- [ ] exception
- [ ] throws
- [ ] raise

Which of the following statements about user-defined exceptions in Java is correct? *

- [x] User-defined exceptions must always extend the built-in Exception class.
- [ ] User-defined exceptions can be created by simply declaring a new class with any name.
- [x] User-defined exceptions can only be thrown explicitly using the throw keyword.
- [ ] User-defined exceptions must always extend the built-in RuntimeException class.

*

```
class Demo1{ }
class Demo2 extends Demo1
{

}
public class Main
{
    public static void main(String[] args){
    Demo2 d = new Demo1();
}
}
```

- ○ is perfectly valid.
- ◉ 1Error: Incompatible data type
- ○ is valid using abstract class
- ○ none of the above
- ○ Other:

What are generic methods?

- ○ Generic methods are the methods defined in a generic class
- ○ Generic methods are the methods that extend generic class methods
- ◉ Generic methods are methods that introduce their own type parameters
- ○ Generic methods are methods that take void parameters

Clear selection

An abstract class is declared using the keyword abstract that *

- ○ must contain abstract method
- ○ cannot contain abstract method
- ◉ may or may not contain abstract method
- ○ none of the above

The getChars method is used to copy characters from this string into the destination character array. The signature of the method getChars is

- ◉ void getChars(int srcBegin, int srcEnd, char[ ] dst, int dstBebin)
- ○ int getChars(int srcBegin, int srcEnd, char[ ] dst, int dstBebin)
- ○ boolean getChars(int srcBegin, int srcEnd, char[ ] dst, int dstBebin)
- ○ None of the above
- ○ Other:

Clear selection

What will be the output of the following code?

```
class BaseClass
{
        protected final void getInfo()
        {
                System.out.println("Base class");
        }
}

public class ChildClass extends BaseClass
{
        protected final void getInfo()
        {
                System.out.println("Child class");
        }
        public static void main(String[] args)
        {
                BaseClass obj = new BaseClass();
                obj.getInfo();
        }
}
```

- (●) Base Class
- ( ) Child class
- ( ) Runtime Error
- ( ) Compilation Error
- ( ) Other:

Clear selection

What will be the output of following code?

```
// Main.java
public class Main
{
        public static void getString(String s)
        {
                System.out.println("String");
        }
        public static void getString(Object o)
        {
                System.out.println("Object");
        }
        public static void main(String args[])
        {
                getString(null);
        }
} //end class
```

- ○ NullPointerException
- ◉ Object
- ○ String
- ○ Compilation Error

Clear selection

Which of the following is an invalid Java generics syntax?

- ○ ArrayList<String> al1 = new ArrayList<String>();
- ○ List<String> al2 = new ArrayList<String>();
- ○ Collection<String> al3 = new ArrayList<String>();
- ◉ ArrayList<Object> al4 = new ArrayList<String>();

Clear selection

class Test<T extends Number> { }

For the below Test class which of the following way of object creation is invalid?

- ○ Test<Integer> t1 = new Test<Integer>();
- ◉ Test<String> t2 = new Test<String>();
- ○ Test<Double> t3 = new Test<Double>();
- ○ Test<Number> t4 = new Test<Number>();

Clear selection

Choose the correct option based on this program

```
import java.util.*;
class UtilitiesTest {
    public static void main(String[] args) {
        List < int > intList = new ArrayList < > ();
        intList.add(10);
        intList.add(20);
        System.out.println("The list is: " + intList);
```

- ○ It results in a runtime exception
- ◉ It prints the following: The list is: [10, 20]
- ○ It prints the following: The list is: [20, 10]
- ○ It results in a compiler error

Clear selection

What will be the output of the below ArrayList program?

```
import java.util.ArrayList;
import java.util.List;

public class Test {
   public static void main(String[] args) {
      List<String> al = new ArrayList<String>();
      al.add("A");
      al.add("B");
      al.add("A");
      al.add(null);
      System.out.println(al);
   }

}
```

- (●) [A, B, null]
- ( ) Compile time error
- ( ) [A, B, A, null]
- ( ) [A, B]

Clear selection

Which is the root interface of the Java Collection framework hierarchy?

- ( ) Root
- (●) Collection
- ( ) List/Set
- ( ) Collections

Clear selection

Find the output of the above Test class if we develop MyComparator class as below?

```
class MyComparator implements Comparator<Integer> {
    @Override
    public int compare(Integer i1, Integer i2) {
        if (i1 < i2) return +1;
        else if (i1 > i2) return -1;
        else return 0;
    }
}
```

- ○ [100, 0, 150, 50, 125, 125]
- ○ [100, 0, 150, 50, 125]
- ◉ [150, 125, 100, 50, 0]
- ○ [0, 50, 100, 125, 150]

Clear selection

Predict the output of following code *

```
class Test
{
    int i;
    Test(int i)
    {
        this.i=i;
    }
    void display()
    {
        System.out.println(i);
     }
}

public class Main
{
    public static void main(String[] args)
    {
        Test t1 = new Test(5);
        Test t2= new Test();
        t1.display();
        t2.display();
        }
}
```

- ○ Garbage Value
- ○ 5
- ◉ Compilation Error
- ○ Code will compile successfully, but will not execute

A copy constructor is a constructor in which _____ object is passed as an argument. *

- (●) One
- ( ) Two
- ( ) None
- ( ) N number

Which of the statements is correct for function overloading? *

- (●) Return type does not serve any purpose
- ( ) Type of argument should be equal for the function
- ( ) Number of arguments should be same
- ( ) All of the above

int arr[3][2][2]={1,2,3,4,5,6,7,8,9,10,11,12}; *
What values does arr[2][1][0] in the sample code above contains?

- ( ) 5
- (●) 9
- ( ) 7
- ( ) 11

*

In the Chain of Responsibility design pattern, which of the following statements is true?

A) Each handler in the chain must handle the request regardless of its type.
B) The request is processed by multiple handlers simultaneously.
C) The chain can only have a single handler.
D) The chain of handlers is fixed and cannot be modified at runtime.

- (●) A
- ( ) B
- ( ) C
- ( ) D

*

Match the pairs.

Elements:	Descriptions:
1)Pattern Name	A. Describes the specific problem or scenario that the pattern addresses.
2)Problem	B. Provides a unique identifier for the pattern and helps in communication and referencing.
3)Solution	C. Outlines the trade-offs and implications of using the pattern.
4)Consequences	D. Presents the design solution or approach to solve the problem.

- ( ) 1-C, 2-A, 3-D. 4-B
- (●) 1-B, 2-A, 3-D. 4-C
- ( ) 1-B, 2-C, 3-DA 4-D
- ( ) 1-C 2-A, 3-B. 4-D

*

What are the different characteristics of design patterns? Select the correct option.

A) Design patterns provide proven solutions for recurring design problems.
B) Design patterns are language-specific and cannot be applied across different programming languages.
C) Design patterns promote code reusability and maintainability.
D) Design patterns enhance communication and understanding among software developers.

- (●) A,B,C,
- ( ) B,C,D
- ( ) A,B,D
- ( ) A,C,D

*

Understand the Analogy of the Keeper and select the correct Design Pattern it identifies to.

Imagine a high-security vault in a bank that stores precious gems. The vault has only one key, and only one person can access it at a time. This person, known as the "Keeper," has the exclusive responsibility of managing the vault and its contents. Whenever someone needs to retrieve or deposit a gem, they must go through the Keeper.

Options:

A) Singleton Pattern
B) Bridge Pattern
C) Chain of Responsibility Pattern
D) Abstract Factory Pattern

- ○ A
- ◉ B
- ○ C
- ○ D

*

Understand the Analogy of the Customer Support System and select the correct Design Pattern it identifies to.

Consider a customer support system in a company where customer queries and issues are addressed. The customer support team is organized in a hierarchical structure, similar to a sequence. Each member of the team has a specific role and responsibility.

When a customer submits a request or complaint, it goes through the sequence of support representatives. The first representative in the sequence receives the request and assesses if they can handle it. If they can, they resolve the issue and the process ends there. However, if the representative cannot handle the request, they pass it along to the next person in the sequence who has more expertise in that specific area. This process continues until the request is either resolved or reaches the end of the sequence

A. Singleton Pattern
B. Bridge Pattern
C. Chain of Responsibility Pattern
D. Abstract Factory Pattern

- ○ A
- ○ B
- ◉ C
- ○ D

*

Which of the following is NOT types of Structural Patterns

1. Adapter Pattern
2. Bridge Pattern
3. Composite Pattern
4. Decorator Pattern
5. Facade Pattern

Options:

A) 1
B) 2
C) 3
D) 4
E) None of the above

- (●) A
- ( ) B
- ( ) C
- ( ) D
- ( ) E

*

A potential downside of the Chain of Responsibility pattern is that a request may go Unhandled or Undelivered if there's no handler capable of handling it, resulting in an unhandled request.
True or False.

- (●) The statement is True
- ( ) The statement is False
- ( ) The statement is ambiguous.
- ( ) None of the above

Submit Clear form