**Sir Syed University of Engineering & Technology**

ANSWER SCRIPT

|  |  |
| --- | --- |
| Date: | 15th June , 2021 |
| Roll Number: | SE-20F-022 |
| Section: | A |
| Name: | Abdul Moiz Chishti |
| Course Name: | Object oriented programming |
| Degree Program: | Software Engineering |
| Total number of pages being submitted: |  |

**Ques 1:**

a) To create a String object, you must use the keyword new and explicitly call the class

constructor

b) When you compare Strings with the == operator, you are comparing their memory

addresses, not their values.

c) When you compare Strings with the equals () method, you are comparing their values,

not their memory addresses

JUSTIFICATIONS:

The equals() method compares two strings, and returns true if the strings are equal, and false if not. In simple words, == checks if both objects point to the same memory location whereas. equals() evaluates to the comparison of values in the objects

Coding example:

public class Test {

public static void main(String args[]) {

Integer x = 5;

Integer y = 10;

Integer z =5;

Short a = 5;

System.out.println(x.equals(y));

System.out.println(x.equals(z));

System.out.println(x.equals(a));

}

}

a. You use the keyword inherits to achieve inheritance in Java.

b. A derived class can access directly all its parents’ nonprivate methods.

c. Subclasses are more specific than the superclass they extend

JUSTIFICATIONS:

A subclass is more specific than its superclass because it represents a smaller group of objects as subclass is a class that derives from another class and inherits state and behavior from all of its ancestors.

Coding example:

class Animal {

String name;

public void play() {

System.out.println("I like to play");

}

}

class Dog extends Animal {

public void display() {

System.out.println("My name is " + name);

}

}

class Main {

public static void main(String[] args) {

Dog labrador = new Dog();

labrador.name = "Scar";

labrador.display();

labrador.play();

}

**Ques 2:**

package paper2;

public class Paper2 {

public static void main(String[] args) {

EducationInsurance a=new EducationInsurance("Abdul Moiz Chishti",001,50000,"15 years");

a.annuallyPaymentPlan();

}}

package paper2;

public interface FixedInsurance {

final String beginner="5 years";

final String intermediate="10 years";

final String advance="20 years";

}

package paper2;

public abstract class Insurance implements FixedInsurance {

String customer\_name,timeperiod;

int policynumber, insurance\_amount, interestrate;

public Insurance(String customer\_name,int policynumber,int insurance\_amount,String timeperiod){

if(insurance\_amount>5000000)

System.out.println("Amount Invalid");

else

{

this.customer\_name=customer\_name;

this.insurance\_amount=insurance\_amount;

this.policynumber=policynumber;

this.timeperiod=timeperiod;

}

}

public void toSting(){

System.out.println("Customer Name : "+ customer\_name);

System.out.println("Insurance Amount : "+insurance\_amount);

System.out.println("Policy Number : "+ policynumber);

System.out.println("Time Period : "+timeperiod);

System.out.println("Interest Rate : "+interestrate+" per year");

}

abstract void annuallyPaymentPlan();

}

class EducationInsurance extends Insurance{

public EducationInsurance(String customer\_name, int policynumber, int insurance\_amount, String timeperiod) {

super(customer\_name, policynumber, insurance\_amount, timeperiod);

interestrate=(int)(insurance\_amount\*0.015);

}

@Override

void annuallyPaymentPlan() {

String type="Insurance Type : Intermediate";

toString();

}

}

**Ques 3:**

If the exception occurs statement 4 will be executed as The **finally block** executes whether exception rise or not and whether exception handled or not. Same as statement 5 will be executed as it is out of try and catch block..

(c) If the line 8 is replaced by catch(Exception e) it will give an error as there is already a catch block so there is no use of it.

**Programming Part:**

package javaapplication38;

class calculator {

int a,b;

calculator(){

};

calculator(int a,int b){

this.a=a;

this.b=b;

}

public void division() throws ArithmeticException{

if(b<0) {

System.out.println("answer is infinity");

throw new ArithmeticException();

}//if

else {

System.out.println("Division is : "+(a/b));

}

}

}

public class JavaApplication38 {

public static void main(String[] args) {

calculator ob=new calculator(36,2);

ob.division();

calculator ob1=new calculator(10,-3);

ob1.division();

}

}

**Ques 4:**

package book;

/\*\*

\*

\* @author Abdul Moiz Chishti

\*/

import java.util.Scanner;

public class Book {

String bn;

String py;

String b\_name;

String b\_writers;

String b\_printer;

String b\_barcode;

String b\_cost;

String b\_publicationyear;

String b\_copiesavailable;

Book(String name,String writers,String printer,String barcode,String cost,String py,String ca){

b\_name=name;

b\_writers=writers;

b\_printer=printer;

b\_barcode=barcode;

b\_cost=cost;

b\_publicationyear=py;

b\_copiesavailable=ca;

}

void Display(){

System.out.println(b\_name);

System.out.println(b\_writers);

System.out.println(b\_printer);

System.out.println(b\_barcode);

System.out.println(b\_cost);

System.out.println(b\_publicationyear);

System.out.println(b\_copiesavailable);

}

String barcode;String cost;String writers;

public void getbyprinters(String barcode,String cost,String writers){

this.barcode=barcode;

this.cost=cost;

this.writers=writers;

}

public void setbyprinters(){

System.out.println("Barcode :"+barcode);

System.out.println("Cost :"+cost);

System.out.println("Writers :"+writers);

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

Book myobj=new Book("The Kite Runner","anonymous","anonymous","2345","1250/-","2002","19998");

myobj.Display();

myobj.getbyprinters("8992943", "899", "J.K Wroling");

myobj.setbyprinters();