**OOP Lab 6**

|  |  |  |  |
| --- | --- | --- | --- |
| Name: |  | Department: |  |
| Student ID: |  | Room Number: |  |
| Due Date: | **April 13, 23 : 59** | | |

* **Submit your assignment using the following file format:**

LabNumber\_Name\_IdNo.zip (eg. Lab6\_Hongkildong\_2017.zip).

* This zip file will contain **two types of** files, namely:

1. **report file** with file format **“Report\_Lab number**” (eg. report\_6) to answer theory questions and to write the screen shot of your program.
2. Source code file that contains codes of classes to answer programming questions.

**Contents**

1. **Objec**t class defines the following most common methods shared by any class in java.
2. public **String** **toString**()
3. public native int **hashCod**e()
4. public boolean **equals(Object** o)
5. public final Class **getClass**()
6. **Class** class, **String** class, **ArrayList** Class and **Integer**(Wrapper Class) are also covered

**Problem 1(): Use of “Class” class and reflection API.**

**// ObjectTest.java**

L1: **import java.lang.reflect.\*;**

L2: public c**lass ObjectTest**

L3: {

L4: public static void **main** (String []args )**throws Exception**

L5: {

L6: int count=0;

**L7: Class** c = Class.**forName** (“**java.lang.Object**”);

L8: **Method**[]m = c.getDeclaredMethods();

L9: **for**( **Method** mi:M ) //**java.lang.reflect.Method**

L10: {

L11: System.out.println (mi.getName());

L12: counter++;

L13. }

L14: System.out.println (“the number of methods:” + count);

L15: **}// end of main**

L16: }// end of class

**Question**: If you delete line 1, what happens? (1pt)

**Problem 2:** **toString** () method of “**Object**” class ()

Question(a): run the following code and take the screenshot (1pt)

**// Student.java**

L1: class **Student**

L2: {

L3: String name;

L4: int ID;

L5: public **Student**(String name, int ID)

L6: {

L7: this.name=name;

L8: this.ID=ID;

L9: }

**L10: public String toString()**

**L11: {**

**L12: return “Student name:” + name + ”and ID:” + ID;**

**L13: }**

L14: **public** static void main (String []args )

L15: {

L16: Student S1 = new Student(“kim”, 101);

L17: Student S2 = new Student (“Homin”, 102);

L18: System.out.println (S1); **/\* 1 \*/**

L19: System.out.println (S1.toString()); /\* **2** \*/

L20: System.out.println (S2); /\* **3** \*/

L21: }

L22:}

**Question(b): From Question(a), comment Line 10-13 and run the program. Show the screen shoot. Why the screen shot is different from Question (a)? Explain your answer. Hint: the** standard implementation of **toString()** method in the **Object** class is as follows (3pt).

**L1: public String toString()**

**L2: {**

**L3: return getclass().getName()+ ”@” +Integer.toHexString(hashCode());**

**L4: }**

// a) getclass(): method of “Object” class

// b)getName(): method of “Class” class

// c)Integer.toHexString(): method of Integer wrapper class

// d)hashCode(): method of the “**Object**” Class

**Problem 3: toString()and hashCode() methods of Object class**

**// Test.java**

L0: import java.util.\*;

L1: **public class Test**

L2: {

**L3:**

**L4: public** static void main (String []args )

L5: {

L6: **String** **s** =new String (“kim”);// built-in **String** class

L7: System.out.println (**S**);

L8: **Integer** **I** =new Integer (10);// built-in **wrapper** class

L9: System.out.println (**I**);

L10 **ArrayList** L =new **ArrayList** (); // built-in **collection** Class

L11: L.**add**(“A”);

L12 L.**add**(“B”);

L13: System.out.println (**L**);

L14: **Test** **t** =new Test();// **user defined class**

L15: System.out.println (**t**);

L16: }

L17 }

Question(a): Is java.lang.**String** class **override** **toString**() method of “Object” class? Explain your reason. Hint: refer line 7 (**2pt)**

Question(b): Is java.lang.**Integer** class **override** **toString**() method of “Object” class? Explain your reason. Hint: refer line 9(**2pt**)

Quetsion(c): Is java.util.**ArrayList** class **override** **toStrin**g() method of “Object” class? Explain your reason. Hint: refer line 13(**2pt)**

Question(d) :Is **Test** class **override** **toString**() method of “**Object**” class? Explain your reason. Hint: refer line 14(**2pt**)

**Program 4: Overriding toString()method by Overriding only hashcode()method**

**// Student.java**

L1: **class Test**

L2: {

L3: **int i;**

L4:

L5: public **Test** (int i);

L6: {

L7: this.i=i;

L8: }

**L9**: **public int hashCode().**

**L10: {**

**L11: return i;**

**L12: }**

**L13: public String toString().**

**L14: {**

**L15: return** **i** **+** ” “;

**L16: }**

L17: public **static** void main (String []args)

L18: {

L19: Test t1 = new Test (10);

L20: Test t2 = new Test (100);

**L21: System.out.println (t1);**

**L22: System.out.println (t2);**

L23: }

L24: }

1. comment Line **9-16** and see the output. Why you get this output?(**3pt**)
2. commnet Line **13-16** and see the output. Why you get this output?(**3p**t)
3. remove the comment **9-16** and see the result. Why you get this output?(**3pt**)

**Problem 5:** Difference between operator (= =) and **equals ()** method in “**Object**” Class

Remark 1: Assume that the meaning of equality is to comparing only names, only roll numbers or both.

Remark 2: When we pass heterogeneous objects, avoid rising of Class Cast exception

Remark 3: When we pass null argument, avoid rising of Null Pointer Exception

1. **Run the following Code and take screen shoot (1pt).**

**// Student.java**

L1: class **Student**

L2: {

L3: **String** name;

L4: **int** ID;

L5: public **Student** (String name, int ID)

L6: {

L7: this.name=name;

L8: this.ID=ID;

L9: }

**L10: public boolean equals(Object obj)// S1.equal(S2)**

**L11: {**

**L12: try**

**L13: {**

**L14 String name1 =this.name; // this refers to S1.**

**L15: int ID1=this.ID;**

**L16; Student s =(student)Obj;**

**L17: String name2 =s.name;**

**L18: int ID2 =S.ID;**

**L19: if( name1.equals(name2)&& (ID1==ID2))**

**L20: return true**

**L21: else**

**L22: return false;**

**L23: }// end of try block**

**L24: Catch(** **ClassCastException e)**

**L25: {**

**L26:** return false;

**L27: }**

**L28 Catch(** NullPointer**Exception e)**

**L29: {**

**L30:** return false

**L31: }**

**L32: } // end of equals()**

L33: **public** static void main (String []args )

L34: {

L35: **Student S1 = new Student(“kim”, 101);**

L36: Student S2 = new Student (“Homin”, 102);

L37: **Student S3 = new Student (Homin, 102);**

L38: **Student S4 = S1**

L39: System.out.println ( **S1.equals(S2**));

L40: **System.out.println ( S1.equals(S3));**

L41: System.out.println ( **S1.equals(S4**));

**L42: System.out.println ( S1.equals(“kim”));**

**L43 System.out.println ( S1.equals(null));**

**L44: } end of main**

**L45: }// end of class**

1. **In the above code, comment Line 10-32 and comment lines 42-43.**

**What is the output at line 39, 40 and 41? Why you get this output (3pt).**

**c) In the above code, comment only Line 10-32. Is there error at line 42 and Line 43? Why? (3pt).**

**Problem 6: Replace the line 10-23 in Problem 5 by the following code.**

**Did you get the same result as in (a) in the problem 5? Explain your reason (3pt)**

**L0 Public boolean equals(Object obj)// S1.equal(S2)**

**L1: {**

**L2: try**

**L3: {**

**L4; Student s =(student)Obj;**

**L5: if( name.equals(s.name)&& (ID==S.ID) )**

**L6: return true**

**L7: else**

**L8: return false;**

**L9: }// end of try block**

**L10: Catch(** **ClassCastException e)**

**L11: {**

**L12:** return false;

**L13: }**

**L14 Catch(** NullPointer**Exception e)**

**L15: {**

**L16:** return false

**L17: }**

**L18: }// end of equals()**

**Problem 7: Replace the line 10-23 in Problem 5 by the following code.**

**Did you get the same result as in (a) in problem 5? Explain your reason (4pt)**

**L1: public boolean equals(Object obj) // S1.equal(S2)**

**L2: {**

**L3; if(obj instanceof Student)**

**L4: {**

**L5: Student s =(student)Obj;**

**L6: if( name.equals(s.name)&& (ID==S.ID) )**

**L7: {**

**L8: return true**

**L9: }**

**L10: else**

**L11 {**

**L12: return false;**

**L13: }**

**L14: }// end of if block**

**L15: else**

**L16: {**

**L17 return false;**

**L18: }**

**L19: }**  **end of equals()**

**Problem 8: Replace the line 10-23 in Problem 5 the following code.**

**Did you get the same result as in (a) in problem 5? Explain your reason (4pt)**

**L1: public boolean equals(Object obj) // S1.equal(S2)**

**L2: {**

**L3: if(obj == this)// instead of comparing field by field**

**L4: {**

**L5: return true;**

**L6: }**

**L7; if(obj instanceof Student)**

**L8: {**

**L9: Student s =(student)Obj;**

**L10: if( name.equals(s.name)&& (ID==S.ID) )**

**L11: {**

**L12: return true**

**L13: }**

**L14: else**

**L15 {**

**L16: return false;**

**L17: }**

**L18: }// end of if block**

**L19: else**

**L20: {**

**L21 return false;**

**L22: }**

**L23: }**  **end of equals()**