CSE 2102: Object Oriented Programming
2nd Year 1st Semester 2018
Quiz 01 (25 Marks)
Time: 1 hour

B

1. Differentiate between method overloading and method overriding using appropriate **code only**. [2]
2. Consider the following code: [4]

```
interface Customer
{
     public void setCustid(int custid);
     public int getCustid();
     public void setCustName(String custname);
     public String getCustName();
     public void DisplayCustomer();
}
```

   Write a *CustImplement* class by implementing the above interface Customer. The class *CusImplement* has two data members *custid* and *custname*. Provide the parameterized constructor for the class *CustImplement*. You don't need to write the main method.
3. Write True/False: [5]
   i. To sort an array in a method, both it and its size must be passed in as parameters.
   ii. Every constructor must have void in place of a return type because a constructor cannot return a value.
   iii. Under inheritance, a superclass inherits all of the members in all of its subclasses.
   iv. The principle of overloading is what makes polymorphism work in Java.
   v. A method inside an abstract class must be declared abstract.
4. You are asked to model an application for storing data about people. You should be able to have a person and a child. The child is derived of the person. Your task is to model the application. [4]
   - Person – represents the base class by which all others are implemented
     - People should not be able to have negative age
   - Child - represents a class which is derived by the class Person.
     - Children should not be able to have age greater than 15

   Constraints
   - If the age of a person is negative –message is: "Age must be positive!"
   - If the age of a child is bigger than 15 –message is: " Child's age must be lesser than 15!"
   - If the name of a child or a person is no longer than 3 symbols – message is: "Name's length should not be less than 3 symbols!"
   - Write a toString() method to print the contents (name, age, height) of a specific object.
5. [2]

```
class A{
     int i;
}
class  B extends A{
     int i;
     void test(){
             int i;
               //add your code here
     }
}
```

   Inside the test() method, add code so that the following things are achieved-
   i) Value of i of class A becomes 10,
   ii) Value of i of method test() becomes 20,
   iii) Value of i of class B becomes 30.
6. What is the output of the code below? If there is an error, state the problem here. [2]

```
package com;
class Animal {
   public void printName(){
     System.out.println("Animal");
   }
}
package exam;
import com.Animal;
public class Cat extends Animal {
   public void printName(){
     System.out.println("Cat");
   }
}
```

CSE 2102: Object Oriented Programming
2nd Year 1st Semester 2018
Quiz 01 (25 Marks)
Time: 1 hour

```
package exam;
import com.Animal;
public class Test {
   public static void main(String[] args){
      Animal a = new Cat();
       a.printName();
   }
}
```

7. What is the output of the code below? If there is an error, state the incorrect line's number. [2]

```
public class A {
   public void printName(){
System.out.println("Value-A");}
}
public class B extends A{
   public void printName(){
      System.out.println("Name-
B");}
}
public class C extends A{
   public void printName(){
      System.out.println("Name-
C");}
}
```

```
1. public class Test{
2.   public static void main(String[]
args){
3.      B b = new B();
4.      C c = new C();
5.      b = c;
6.      newPrint(b);
7.    }
8.    public static void newPrint(A a){
9.      a.printName();
10.   }
11.}
```

8. What is the output of the code below? If there is an error, state the incorrect line's number. [2]

```
1. public interface InfA {
2.       protected String getName();
3. }
4. public class Test implements InfA{
5.       public String getName(){
6.           return "test-name";
7.       }
8.       public static void main (String[] args){
9.           Test t = new Test();
10.          System.out.println(t.getName());
11.      }
12. }
```

9. Given: [2]

```
1. interface Horse { public void nicker(); }
```

Which will compile? (Choose all that apply.)

A.
```
public class Eyra implements Horse { public void nicker() { } }
```
B.
```
public class Eyra implements Horse { public void nicker(int x) { } }
```
C.
```
public class Eyra implements Horse {
   public void nicker() { System.out.println("huhuhuhuh..."); }
}
```
D.
```
public abstract class Eyra implements Horse {
   public void nicker(int loud) { }
}
```
E.
```
public abstract class Eyra implements Horse {
   public void nicker(int loud) ;
}
```