Bangladesh Army University of Science and Technology

Department of Computer Science and Engineering

Final Examination, Fall 2018 | **Level-2 Term-II**
Course Code: CSE 2203 | **Course Title: Object Oriented Programming II**
Time: 03 (Three) hours | **Full Marks: 210**

N.B. (i) Answer any three questions from each PART (ii) Use separate answer script for each PART
(iii) Marks allotted are indicated in the margin (iv) Symbols have their usual meanings

PART A

(Answer any **three** questions)

1. a) What is the signature of Java main method? Who invoke the main method? Briefly describe the significance of each term. **5**

   b) What will happen if we declare a variable, a method, and a class as final? The following program contains some bugs regarding final keywords. Debug the program and explain the cause of bugs. **(4+10) =14**

```
final class A {
    final int FILE=1;
    A(int f){this.FILE=f;}
    final void show(){System.out.println(FILE);}
}
class B extends A{
    void show(){ System.out.println("Nothing");}
}
```

```
class Final{
        public static void main(String args[]){
                A a1= new A(2);
                A a2 = new B();
                a.show();
        }
}
```

   c) Consider the following java classes. **(2x8) =16**

```
class A{
  A(){ System.out.println("DC-A "); }
  A(String n){ System.out.println(" PC A. "+n); }
  void show(){ System.out.println(" Hello A ");}
  void show(String n){
      System.out.println(" Hello  A P " + n );
  }
}
```

```
class B extends A{
 B(){ System.out.println(" DC-B "); }
B(String n){ System.out.println(" PC B. "+n); }
void show(){ System.out.println(" Hello B "); }
 void show(String n){
        System.out.println(" Hello B P " + n );
 }
}
```

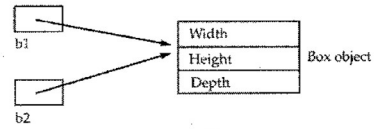
   Now, identify the following process (if any) and also state how many times the scenario occurs in the program.

Default Constructor	*Inheritance*	*Encapsulations*	*Overriding*
Parameterized Constructor	*Destructor*	*Object/Instance*	*Overloading*

2. a) **(5+5) =10**

   "*Obtaining an Object in java is a two-step process*"- explain with example.

   Write the java code for the following scenario.



   b) Write the usage of instance and class variable. Write the restrictions of static method. Give the output of the following code segment? What will happen if you remove the *static* keyword from any of the *sheep_talk* or *kid_talk* method? Why? **(5+5+5) =15**

```
static  void kid_talk() {
        System.out.println("Maaa...");
        sheep_talk();
}
static  void sheep_talk() {
        System.out.println("Baaa...");
        kid_talk ();
}
```

```
static{   kid_talk(); }
static void init(String st){
    System.out.println(" st  ");
}
public static void main(String args[])
{   System.out.println("Animal  Talking Habit ... ");
    init("This is trap");
}
```

   c) Consider the following scenario: **10**

   Students of a class are identified by id and also have name and cgpa as attributes. Every student has a department attribute of type string and all the objects once assigned get the same department. Provide necessary constructors and a main method. Write a complete java program to create objects and show interactions.

3. a) How to access parent class members (variables, methods constructors) using super? How can a superclass reference be used to access the members of subclass? Explain both with example. **(8+7) =15**

b) Consider the following code segment. Write the Java function minmax. You can write only one minmax function. **10**

```
public static void main (String  args[]) {
        int a = minmax ("min", 2, 1,6,4,5);  //a=1
        int b = min max ("min", 3, 0, 6);     //b = 0
        int c =minmax ("max", 1,2,6,5);    // c = 6
        int d.=minmax ("max", 1 ,3,7);     // d = 7
}
```

c) Define ambiguity in varargs with a suitable example. Which of the following varargs prototypes are perfectly valid and which are not? If you find any invalid prototype write them in correct form. **10**

(i)vaTest(int ...v, float f, double ...d); (ii). vaTest(int ...v, boolean...b); (iii). vaTest(int a, int b, int...v);

(iv). vaTest(String st, boolean...b); (v). vaTest(int i, float f, double ...d, boolean b);

4. a) How does interface in Java enable multiple inheritances? Consider the following code segment. Write minimum code in class C2 for successful compilation. You can't define C2 as abstract. **(5+5) =10**

```
interface I1 {
default void f1(){ ... ...}
        void f2();
}
interface I2{
     void f3();
     void f4();
}
```

```
abstract class  C1  implements I1 {
        abstract void f5();
        final void f6() {... ... }
}
class C2 extends C1 implements I2 {
        // ... ... ...
}
```

b) Consider the following java code segment **10**

```
class Base{
    public Base(){}
    public abstract void display(){}
}
```

```
class Child extends Base{
    public Child(){ super();}
    public abstract void display(){}
}
```

Answer the following:

i. What is wrong in the above code? Explain briefly.
ii. Propose necessary changes in the above code for your expected output.

c) "Java static property share to all object" – explain. Consider the following code segment. How can you access volume(), area() and dim() methods from main method? **15**

```
class  Box1{
    static  void volume(){
        System.out.println("Volume" );
    }
}

class  Box2{
    void dim(){
        System.out.println("Dimention" );
    }
}
```

```
class Access extends Box2{
static void show(){
        System.out.println("Area");
    }
public static void main(String args[]) {
        // ... ... ...
    }
}
```

PART B

(Answer any **three** questions)

5. a) What is a thread? Why thread scheduler is needed? What are the differences between preemptive scheduling and time slicing? Write a few lines on thread priority, priority constant and its value. **(2+4+4) =10**

b) Write two different ways to create threads in Java. Which one is better and why? **10**

c) Define *Daemon Thread* with example? How is it different from user thread? What are the methods of daemon thread? **08**

d) What is garbage collection? How to request garbage collector? How can an object be unreferenced and used by garbage collector? **07**

6. a) What do you mean by abstract class? What are the restrictions to classes that extend abstract class? Write the differences between abstract class and interface. Write when an abstract class is essential. **(2+3+5) =10**

b) **10**

What are the members an interface can have? Mention their access protection.

What is the problem with the following code? Write two different ways to fix the problem.

```
interface A {
    default void f() { System.out.println("A's f"); }
}
interface B {
   default void f(){ System.out.printin("B's f"); }
}
public class C implements A, B { ... ... ... }
```

c) Suppose that you develop a java project which contains two packages: P1 and P2. Package P1 contains class A, class B, class C and a sub-package SubP which has a class named SC. Under package P2 there is a public class named D and it include the main function. **(9+6) =15**

Assume that, Class A, B, C and SC contain method *show1(), show2(), show3() and show4()* respectively which all are public and the classes are also public.

Now answer the following questions-

i. How many ways you can access the members of package P1 from package P2? Give example for each of them.

ii. Can you access the resources from sub package by importing the package only? Why or why not? How can you access the member method: *show4()* of class SC from package P2? Give example.

7. a) Why is String objects immutable in java? How to create a mutable String? What is string constant pool? Why is it useful? **12**

b) Following is an example of an incorrect program and will not compile. What are the problems here? How to solve them? Rewrite the program so that it provides the desired outcome. **8**

```
class ThrowsDemo {
    static void throwOne() {
        System.out.println("Inside throwOne.");
        throw new IllegalAccessException("demo");
    }
    public static void main(String args[]) { throwOne(); }
}
```

c) Consider the following instances of String, StringBuilder and StringBuffer classes. Draw the memory mapping (stack, heap and constant pool) for each of them. **10**

1. String st1 ="Dhaka";	4. String st4 =new String("Barisal");
2. String st2 =" Barisal";	5. StringBuffer sb1 =new StringBuffer(" Dhaka ");
3. String st3 ="Dhaka";	6. StringBuffer sb3 =new StringBuffer("Dhaka");
	StringBuilder sb4 =new StringBuilder ("Barisal ");

d) Write the difference between throw and throws keyword. Write when thread is unavoidable. **5**

8. a) Define *functional interface* with example. What is the most common built in *functional interface* in java? **7**

b) Consider the following sample program in java **(3x6) =18**

@FunctionalInterface // Annotation *interface Cab{* *void bookCab();* *}*	*public class LambdaApp{* *public static void main(String args[]){ // Todo code }* *}*

Now, answer the followings:

(i) Develop a class that will implements the *bookCab()* method of the interface.
(ii) How to access the *bookCab()* method using polymorphic statement?
(iii) Give an anonymous class implementation of *bookCab()*.
(iv) How to use lambda expression to implements *bookCab()?*
(v) What will happen if we add another abstract or concrete method in the Cab interface?
(vi) Can we pass any argument in lambda function? If yes, explain how?

c) Write the advantages of generic. What is the problem with the following code segment? How generic can be used to solve this problem as well as individual type casting? **(4+3+3) =10**

```
ArrayList al = new ArrayList();
al.add("Sakib");
al.add("Musfiq");
al.add(10);
```

```
String s1 = (String)al.get(0);
String s2 = (String)al.get(1);
String s3 = (String)al.get(2);
```