

UNIVERSITY OF BARISHAL
Course Title: Object Oriented Programming
Course Code: CSE-2107
2nd Year 1st Semester Final Examination
Admission Session: 2020-2021

Time: 3 Hours

Marks: 60

N.B.: Answer any FIVE questions out of the followings. All parts of each question must be answered consecutively. Right side of the question shows the maximum marks.

- 1.a) Distinguish between the following terms and provide suitable examples for each: 8
- Objects and Classes
 - Data abstraction and data encapsulation
 - Inheritance and polymorphism
 - Dynamic binding and message passing

- b) Find the error in each of the following program segments. Explain how to correct the error. 4

<pre>i) void g() { System.out.println("Inside method g"); void h() { System.out.println("Inside method h"); } }</pre>	<pre>ii) void product() { int a = 6, b = 5, c = 4, result; result = a * b * c; System.out.printf("Result is %d\n", result); return result; }</pre>
<pre>iii) int sum(int x, int y) { int result; result = x + y; }</pre>	<pre>iv) void f(float a); { float a; System.out.println(a); }</pre>

- 2.a) What is a static variable? Explain the static method with a proper example. What is the main method is not public static in Java? 4
- b) What is constructor in Object Oriented Programming (OOP)? Can it possible to write multiple constructors inside a class with different name? Explain your answer. 4
- c) Write down the difference between methods overwrite and overloading. Explain those with example. 4
- 3.a) Suppose you have a Piggie Bank with an initial amount of \$50 and you have to add some more amount to it. Create a class 'AddAmount' with a data member named 'amount' with an initial value of \$50. Now make two constructors of this class as follows: 6
- without any parameter - no amount will be added to the Piggie Bank
 - having a parameter which is the amount that will be added to Piggie Bank
- Create an object of the 'AddAmount' class and display the final amount in Piggie Bank.

- b) What is Encapsulation? Differentiate between an interface and an abstract class with example. 6

- 4.a) Differentiate between Object Oriented Programming and Procedural Programming. 2

- b) Create a Java program that implements an abstract class **Animal** that has a *Name* property of type text and three methods *SetName* (string name), *GetName* and *Eat*. The *Eat* method will be an abstract method of type void. 5

You will also need to create a **Dog** class that implements the above **Animal** class and the *Eat* method that says the 'Dog is Eating'.

To create a new **Dog** type object from the Main of the program using the dog's name, give the Dog object a name, and then execute the *GetName* and *Eat* methods.

- c) Write the code and draw an inheritance hierarchy for students at a university. Use **Student** as the superclass of the hierarchy, and then extend **Student** with classes **UndergraduateStudent** and **GraduateStudent**. Continue to extend the hierarchy as deep (i.e., as many levels) as possible. For example, **Freshman**, **Sophomore**, **Junior** and **Senior** might extend **UndergraduateStudent**, and **DoctoralStudent** and **MastersStudent** might be subclasses of **GraduateStudent**. After drawing the hierarchy, discuss the relationships that exist between the classes. 5
- 5.a) What is JVM? Explain the internal architecture of JVM with neat sketch. 4
- b) What are the available wrapper classes in Java? What is Autoboxing and Unboxing? 4
- c) The annual examination results of 02 (two) students are tabulated as follows: 4

Roll No.	Subject-1	Subject-2

Write a program to read the data and determine the following:

- Total marks obtained by each student.
- The highest marks in each subject and the Roll No. of the student who secured it.

- 6.a) Write down the output of the following programs: 4

```
i) int main()
    { int i=7, j=9, modresult=0,
      divresult=0;
      modresult = i % j;
      cout<< modresult << " ";
      divresult = i / modresult;
      cout<< endl<< divresult;
      return 0;
    }
```

```
ii) int main()
    { int i[5] = { 1, 2, -1, 5}, j;
      (i [1] == i[0]) ? (j=2) (j=1);
      if ( j >= 1)
          cout<< "OK";
      else
          cout<< "YES";
      return 0;
    }
```

- b) Write a code using inheritance to create an exception superclass (called **ExceptionA**) and exception subclasses **ExceptionB** and **ExceptionC**, where **ExceptionB** inherits from **ExceptionA** and **ExceptionC** inherits from **ExceptionB**. Write a program to demonstrate that the catch block for type **ExceptionA** catches exceptions of types **ExceptionB** and **ExceptionC**. 4
- c) Create a custom java package and class for the following problem: 4
- Given the current array of numbers [10, 0, 20, 30, 5, 21, 30], find the largest and smallest value in the unsorted array.
- 7.a) You have three tasks. All of the three tasks need to be done in parallel. In the first task "**TaskA**" you need to output "**Hello World**" 100 times. In second task "**TaskB**" you need to add integers from 1 to 1000, and in the final task "**TaskC**" print "**I can do it**" 200 times and wait for 1 second before writing the line each time. Now ensure that "**TaskC**" will be completed first, then "**TaskA**" and then "**TaskB**". Write down java code for the above problem. 6
- b) Describe the various form of implementing interfaces. Give examples for each case. 6
- 8.a) The Collections class of Java Collection Framework has several methods (e.g. **sort()**, **min()**, **max()**) that require natural ordering of the classes to be handled. Explain the two possible conditions that must be satisfied on the classes to be able to use the **sort()** method to arrange the list of objects these classes. 5
- b) What is a collection framework in Java? Draw the collection framework at a hierarchical level. 3
- c) Distinguish between 4
- InputStream** and **Reader** classes
 - OutputStream** and **Writer** classes

Good Luck!!!