

A.I.S.U.T
Date of Examination: 12/09/18

AISANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Department: Computer Science and Engineering

Program: Bachelor of Science in Computer Science and Engineering

Semester Final Examination: Spring 2018

Year: 1st Semester: 2nd

Course Number: CSE1205

Course Name: Object Oriented Programming

Time: 3 (Three) hours

Full Marks: 70

[There are seven questions carrying a total of 14 marks each. Answer any five questions.

Marks allotted are indicated in the right margin.]

1. a) Describe the three principles of object oriented programming with proper examples. 6
b) What is a class and what is an object? Provide a realistic example of a class. Your class has to have some instance variables and methods. Write down java code for your class. 6
c) How a java program is platform independent? 2
2. a) Consider the java code given below and make improvements of the code in terms of the encapsulation feature of object oriented programming. 5

```
import java.util.Scanner;

class Account{
    String name, accountNo;
    double balance;
    Account(){
        balance = 0.0;
        name = "";
        accountNo = "";
    }
    Account(String name, String accountNo, double balance){
        this.name = name;
        this.accountNo = accountNo;
        this.balance = balance;
    }
}
public class Demo {
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        Account rahimAccount = new Account("Rahim", "1001", 0.0);
```

```

        double depositAmount = sc.nextDouble();
        rahimAccount.balance += depositAmount;
        System.out.println("you have deposited "+depositAmount+", your
balance is "+ rahimAccount.balance);
        double withdrawAmount = sc.nextDouble();
        rahimAccount.balance -= withdrawAmount;
        System.out.println("you have withdrawn "+depositAmount+", your
balance is "+ rahimAccount.balance);
    }
}

```

- b) Suppose that there is an *Employee* class having some attributes: *name (String)*, *eid (int)*, *designation (String)*, *age (int)*. Another class *Employees* has an array that contains *Employee* objects. In the *Employees* class, there should have a functionality (method) to find a particular employee from the array. An employee can be found in three different ways: i) only by a name, ii) only by an eid, and iii) by designation and age (assume that you will find only one employee when considering designation and age). Keep in mind that you can only use "find" as a method name that is related to finding an employee. Design the *Employee* and *Employees* classes by using proper object oriented concept. 7

- c) What are the uses of *this* keyword? 2

- 3 a) Why a constructor is important? Should a programmer use no-argument constructor or a parameterized constructor or both? Provide proper arguments to support your opinion. 5

- b) Suppose that there is a *Student* class that has all the information (including personal, academic and financial) about a student of a university. In the university, there are two departments: computer science and engineering (CSE) and electrical and electronics engineering (EEE). Students from CSE and EEE share all attributes with *Student*. Consider that the *Student* class has two methods: one for calculating cgpa (*calculateCgpa*) of a student, other for calculating semester fees (*calculateSemesterFees*). The method that calculates cgpa is same for all the students, however, the other one is not the same. Students of CSE get 10% discount while EEE students get 15% discount on annual fee which is 10000\$. 9

Design necessary classes for the above scenario in terms of inheritance and abstract method. You do not need to write code for *calculateCgpa* method..

- 4 a) A personal computer has to do several tasks, such as, taking inputs from keyboard, processing the inputs according to an user written program, and display outputs on the monitor. Any personal computer has to perform all the above tasks. For example, it is not possible for a personal computer to take inputs from a keyboard, process it, but not displaying outputs. 8

By considering above scenario, you need to design two personal computer classes (e.g., *AsusComputer*, *DellComputer*). Your classes should have three methods to handle the above described tasks. While designing these classes use an appropriate object oriented design.

b) Consider the following piece of code and assume that the *Patient* is a class having proper constructors and boolean **equals(Object o)** method that is overridden to compare the equalities of the values of two Patient objects:

```
Patient p1 = new Patient ("Rahim", "12-10-1990", 78);
```

```
Patient p2 = new Patient (p1);
```

What are the values of the expression **p1.equals(p2)** and **p1 == p2**? Support your opinion with appropriate arguments.

c) "Java string is called immutable" - explain with examples.

5. a)

```
class Vehicle{  
    void move() {  
        System.out.println("A vehicle is moving");  
    }  
    void stop() {  
        System.out.println("A vehicle is stopped");  
    }  
}
```

```
class Car extends Vehicle{  
    void move(){  
        System.out.println("Move the gear shift into drive position");  
        System.out.println("A car starts to move");  
    }  
  
    void stop() {  
        System.out.println("Push on a brake pedal");  
        System.out.println("The car is stopped");  
    }  
  
    void moveSourceToDestination(String source, String destination) {  
        System.out.println("Starts from source: "+source);  
        move();  
        System.out.println("Approach to the destination: "+destination);  
        stop();  
    }  
}
```

```

class Bus extends Vehicle{
    void move(){
        System.out.println("Press down fully on the clutch");
        System.out.println("Move the gear shift into first gear");
        System.out.println("Slowly lift your foot off the clutch");
        System.out.println("A bus starts to move");
    }

    void stop() {
        System.out.println("Press down fully on the clutch");
        System.out.println("Push on a brake pedal");
        System.out.println("The bus is stopped");
    }

    void takePassenger(String busStopName){
        System.out.println("Approach to a bus stop: "+busStopName);
        stop();
        System.out.println("Take passenger and move");
        move();
    }
}

public class Demo {
    public static void main(String[] args) {
        /*Line # 1*/ //Vehicle v;
        /*Line # 2*/ Car c = new Car();
        /*Line # 3*/ Bus b = new Bus();

        /*Line # 4*/ //v = c;
        /*Line # 5*/ c.moveSourceToDestination("Firmgate", "AUST");
        /*Line # 6*/ //v.moveSourceToDestination("Firmgate", "AUST");

        /*Line # 7*/ System.out.println();

        /*Line # 8*/ //v = b;
        /*Line # 9*/ b.takePassenger("Tejgoan");
        /*Line # 10*/ //v.takePassenger("Tejgoan");
    }
}

```

Write down the output of the java code segment. Please note that line # 2, 3, 5, 7 and 9 are not commented out.

Now, consider that line # 1, 2, 3, 4, 6, 7, 8, and 10 are uncommented while the line # 5 and 9 are commented out. Does the modified code segment produces same output as before? Please provide explanations to support your opinion.

- b) Write a java program that can copy the content of a file located at *c:\test.txt* to another file located at *d:\test.txt*. 3

c) Why a **main()** method is declared as public and static? Please explain. 3

6. a) Why does the following java code generate a compile time error? Provide proper explanations and make proper corrections. Note that an ArithmaticException is thrown when a denominator is given as 0, an InputMismatchException is thrown when a given input does not match with expected input type. 4

```
import java.util.InputMismatchException;
import java.util.Scanner;

class Demo{
    public static void main (String args[]){
        Scanner sc = new Scanner(System.in);
        int a, b;
        double result=0;
        boolean haveResult=true;
        try{
            a = sc.nextInt();
            b = sc.nextInt();
            result = a/b;
        }catch(Exception e){
            System.out.println("Other exceptions have happened.");
            haveResult = false;
        }catch(InputMismatchException e){
            System.out.println("You have not given an integer.");
            haveResult = false;
        }catch(ArithmetricException e){
            System.out.println("An arithmetic exception has happened.");
            haveResult = false;
        }
        if(haveResult)
            System.out.println("Result: "+result);
        sc.close();
    }
}
```

b) Consider that you are developing a hospital management system. Each patient of the system is an object of the *Patient* class. The *Patient* class has the following attributes: *name*, *address*, *age*, *bloodGroup*, *systolicBloodPressure* and *diastolicBloodPressure*. There are some constraints which are as follows: The age should not be less than 5 years; The range of systolicBloodPressure is in between 40 to 250; And diastolicBloodPressure should be within a range of 30 to 140; There are only eight types of blood groups: A+, O+, B+, AB+, A-, O-, B-, and AB-. 8

Design appropriate user-defined exception/s to handle these unexpected situations.
You should have a main method that takes input and throw appropriate exceptions.
Write down the task of a *finally* block?

2

7. a) What is multithreaded programming? What are the advantages of multithreading? 3
 b) Please describe the life cycle of a thread in java. 4
 c) Write a program that will create two threads named "*Even*" and "*Odd*" from main thread. *Even* named thread prints the even numbers within the range from 1 to 100 (format: Even_EvenNumbers, e.g., Even_2, Even_4, ..). On the other hand, *Odd* named thread prints the odd numbers within same range defined above. Main threads should wait until the *Even* and *Odd* threads finish the jobs. 7

Date of Examination: 16/05/2019

AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Department: Computer Science and Engineering

Program: Bachelor of Science in Computer Science and Engineering

Semester Final Examination: Fall 2018

Year: 1st Semester: 2nd

Course Number: CSE1205

Course Name: Object Oriented Programming

Time: 3 (Three) hours

Full Marks: 70

[There are seven questions carrying a total of 14 marks each. **Question # 1 is mandatory.**

Answer any four from the rest.

Marks allotted are indicated in the right margin.]

1. Consider that there is a *Person* class that possesses all the attributes of a person such as name (String), age (int). The classes *Student* and *Teacher* obtains all the persons' attributes. A student has its own attributes such as id (String), department (String), cgpa (double); A Teacher has some attributes such as salary (double), courseTought (Array of String). Moreover, there are four more student classes: *CSEStudent*, *EEEStudent*, *CEStudent*, and *ARCHStudent*. [14]

Code class contains three attributes, *noOfVariables*, *noOfMethods* and *code* (String). Design has the attributes *noOfBuilding* (int) and *design* (two-dimensional int array).

The system designer wants the following facilities:

- I. There should be a system in place so that any person can send and receive a message (String) to/from any other person.
- II. CSE or EEE students can send and receive a code (Code object) to/from any other CSE or EEE students.
- III. CE and ARCH students can send and receive a design (Design object) within themselves.
- IV. Teachers can send and receive code and design within themslves.

Draw a UML class diagram by considering the scenario mentioned above. Your design should be scalable i.e., the system designer should be able to add some more Student class in the future. You should consider that class names bear common meanings regarding super class and sub class relationship.

2. Read the scenario described in Question # 1 and write down all the necessary components in Java by considering the following considerations: [14]
- I. If a teacher X is sending a message to a teacher Y, your program prints two lines (one line for a sender and another line for a receiver): *Teacher X is sending a message to teacher Y. Teacher Y has received the message.*
 - II. If a student X is sending a message to a student Y, your program prints two lines: *Student X is sending a message to Student Y. Student Y has received the message.*

- III. If a student X is sending a code to a student Y, your program prints two lines:
Student X is sending a code to Student Y. Student Y has received the code.
 - IV. If a student X is sending a design to a student Y, your program prints two lines:
Student X is sending a design to Student Y. Student Y has received the design.
 - V. If a teacher X is sending a code to a teacher Y, your program prints two lines:
Teacher X is sending a code to Teacher Y. Teacher Y has received the code.
- You also need to demonstrate the capabilities of your code by using the *main* method.

3. a) Describe the three principles of object oriented programming with proper examples. [5]
- b) The authority of a university wants to save all the information of students, that is, name, id, department, address, mobile number. The ID has four parts: year, department code, session and roll. 04 and 05 are the department of Computer Science and Engineering (CSE) and Textile Engineering (TE) departments, respectively. Rahim and Karim are the students of CSE and TE department, respectively.

Identify classes and objects from the above scenario and write down Java code demonstrating it.

c) class A{ [4]

```
    int a, b;
    A(int a, int b){
        this.a = a;
        this.b = b;
    }
}
```

```
public class TestDemo {
    public static void main(String[] args) {
        /*Line # 1*/ A r, r1, r2;
        /*Line # 2*/ r1 = new A(1,2);
        /*Line # 3*/ r2 = new A(3,4);
        /*Line # 4*/ r = r2;
    }
}
```

Demonstrate graphical representations for each line from line # 1 to line #4 of *TestDemo* class.

4. a) Consider that a programmer wants to design an “Account” class for a bank that has three instance variables, *id*, *name* and *balance*. A customer (e.g., Rahim) can check his/her balance, can deposit and withdraw money. The programmer has written the following code: [5]

```
Class Account {
    String id, name;
    double balance;
}
```

```

Class Demo{
    Public static void main(String args[ ]){
        Account rahimAccount = new Account();

        double depositedAmount = 500;
        rahimAccount.balance += depositAmount;
        System.out.println("current balance: " +rahimAccount.balance);

        double withdrawnAmount = 200;
        rahimAccount.balance -= withdrawnAmount;
        System.out.println("current balance: " +rahimAccount.balance);
    }
}

```

Design a proper encapsulated *Account* class that have all the facilities mentioned above.

- b) A teacher can be defined by the attributes, *name*, *designation* and *department*. A student can have the attributes *name*, *id*, *department*. There is a class named “*Searching*” for searching for a teacher or a student from a Teacher or a Student array by the name attribute. The *Searching* class only contains methods that have the name *doSearch*. No other method name is allowed. Write down the Java code using an appropriate object oriented concept. [7]
 - c) What are the uses of *this* keyword? [2]
5. a) Consider that you are designing a basic calculator that can do addition, subtraction, multiplication and division operations. Moreover, you are going to design a scientific calculator that can do all the computing of basic calculator, and in addition, it can do trigonometric operations, sine, cosine and tangent. Basic calculator prints the results up-to 2 decimal places, whereas scientific calculator prints up-to 4 decimal places. Write Java code for the system using proper object oriented programming concepts. [7]

You might need to following Java methods:

sin x: Math.sin(x)
cos x: Math.cos(x)
tan x: Math.tan(x)

Printing 2 decimal places: System.out.printf("%.2f", value).

- b) Consider that an *Employee* class has the instance variables: *name* (String), *designation* (String), *salary* (double). There is a requirement that printing an object prints *name*, *designation*, and *salary* of the object of the *Employee* class. Write down the *Employee* class with a proper constructor. [4]
 - c) Why a *main()* method is declared as public and static? [3]
6. a) What are the uses of the *throws* keyword? Describe a real-life example where you should use the *throws* keyword. [4]
- b) There is an ATM machine where a person can withdraw money, if he/she has a bank account. What are the exceptions that could happen in the context? [8]

- Design appropriate user-defined exception/s to handle these unexpected situations. Write down the entire code (necessary classes) with the *main* method.
- c) Write down a scenario where *finally* block could be used? [2]
7. a) Which are the two ways that a thread can be created in Java? Why one approach is preferable than the other one? [2]
- b) Please describe the life cycle of a thread in java. [3]
- c) There is a water filter in an office that can contain 5 liters drinking water. The employees of the office can drink water from the filter if there is enough drinkable water. Peons of the office refill the filter. The employees have to wait when no drinking water is available and the peons wait when the filter is full. The employees also notify peons if he/she sees the amount of drinking water is getting low (less than 0.5 liters) after drinking from the filter. [9]

You have to write threads for the employees and the peons of the office. The employees drink random quantity (liters) of water and the peon refills random quantity of water at a time. Use appropriate threading concepts while writing codes.

Ahsanullah University of Science & Technology
Department of Computer Science and Engineering
Year: 1st, Semester: 2nd, Final Examination (Fall 2016)

Course No: CSE 1205
 Full Marks: 70

Course Title: Object Oriented Programming
 Time: 3 Hours

[There are Seven (7) Questions. Answer any Five (5).
 Marks allotted are indicated in the right margin.]

1. a) "Class is logical construct and an object has physical reality" – explain the statement with 3 example.

b) Does Java use call by reference or call by value? – justify your answer with an example. 3

c) Write a program in java to find the maximum digit from a given integer number. 4

Sample Input: 845392

Sample output: 9

d) What will be the output when you attempt to compile and run this code? 4

<pre>class Test1 { Test1(int x[]) { System.out.println("Constructor called "); int i = 0; while (i < 4) { System.out.println(x[i]); i = i + 2; } } }</pre>	<pre>class Test2 { int p[] = {4, 5, 6, 2, 1}; Test1 t1 = new Test1(p); Test2(int i) { t1 = new Test1(p); } public static void main(String[] args) { Test2 t2 = new Test2(5); } }</pre>
---	--

2. a) "A superclass variable can refer a subclass object", do you agree? Justify your answer. 4

b) What do you understand by encapsulation in Java? 4

c) Define the following classes: 6

i. Create a class named **Sphere** having the following instance variables:

-baseRadius(double)

ii. Create another class named **Cube** with the following instance variables:

-length(double)

-width(double)

-height(double)

iii. Create a class **Calculation** having method name calcVolume() overloaded two times.

The first version can calculate the volume of sphere and the second version can calculate the volume of cube. Both methods return the value using following formulas:

Volume of sphere: $(4\pi r^3)/3$

Volume of cube: length * width * height

Sphere and **Cube** classes have corresponding constructor. Both classes have a method name **display()** which prints the value of all the instance variables and volume of corresponding classes. (You have to call the **Calculation** class's overloaded method to print Sphere's and Cube's Volume.)

You need to create two objects for **Sphere** and **Cube**. Use parameterized constructor to initialize instance variables for each object and call the **display()** method.

3.

a)

How can Java achieve multiple inheritance? Explain with example.

3

b)

What is the purpose of garbage collection in Java?

3

c)

Define a class named **ClassA** in a package **packX**. This class will contain two public data member **a** and **b** of double type and an abstract method **compute()**. Now define three classes name **ClassB**, **ClassC** and **ClassD** in packages **packY**, **packZ** and **packW**. Each of the classes is a subclass of **ClassA** and will implement the **compute()** method.

4

The **compute()** method in:

-**ClassB** will compute the maximum of **a** and **b**.

-**ClassC** will compute the minimum of **a** and **b**.

-**ClassD** will compute the result of division(**a/b**).

Now create **Main** class in **packM** package and create one object for each **ClassB**, **ClassC** and **ClassD**. Print out the computed result for each of the object.

d)

What will be the output of following code?

4

<pre>class A { public A() { System.out.println("Class A Constructor"); } } class B extends A { public B() { System.out.println("Class B Constructor"); } }</pre>	<pre>class C extends B { public C() { System.out.println("Class C Constructor"); } } class MainClass { public static void main(String[] args) { C c = new C(); } }</pre>
---	---

a)

"Java String is called immutable" – explain with an example.

2

b)

Write down the differences between interface and abstract class.

3

c)

Write down the prototypes and uses of the following methods:

6

i)**substring()** ii)**concat()** iii)**isAlive()**

- d) What will be the output of following code?

3

```
class Stringoutput {  
    public static void main(String args[]) {  
        StringBuffer s1 = new StringBuffer("Ahsanullah University");  
        StringBuffer s2 = s1.reverse();  
        System.out.println(s2);  
        int start = 10, end = 15;  
        char st[] = new char[end - start];  
        s2.getChars(start, end, st, 0);  
        st[0] = (char)(st[0] + 1);  
        st[0] = (char)(st[3] + 1);  
        System.out.println(st);  
    }  
}
```

5. a) Discuss the difference between runtime and compile time errors. Would you rather have an error discovered at run time or compile time? 3

3

b) Explain **final**, **finally** and **finalize** keywords with an example.

3

c) Write down a code block that generates divide by zero exception and array index out of bound exception and handles them using appropriate exception handler. 4

4

- d) Consider a class **StudentDB** that contains an array of integers named **studentRoll** and array of strings named **studentName**. This class should have a constructor to initialize the instance variables.

4

Create an interface named **StudentSearch** which contains a method **String search(int roll)**. **StudentDB** class implements the **StudentSearch** interface. The **search** method in **StudentDB** receives student roll as parameter and returns the student name.

Now create a **Student** class to take input a student's roll and display the corresponding name of the student.

- a) What do you understand by **thread** synchronization in java? Explain with example. 4

4

- b) If the main thread finishes earlier than the child thread, then the program will misbehave. How can you force that the main thread will be finished last in java? Explain with an example. 5

5

- c) What will be the output of the following code segment? Justify your answer. 5

```
class Threadexample extends Thread {  
    public void run() {  
        System.out.println(Thread.currentThread().getName() + " is running...");  
        Thread t = Thread.currentThread();  
        if (Thread.currentThread().getName().equals("Thread-0")) {  
            System.out.println(t.getName() + ":" + t.getPriority());  
            try {  
                sleep(200);  
            } catch (InterruptedException e) {}  
        }  
    }  
}
```

```

public static void main(String args[]) {
    Thread t1 = new Threadexample();
    Thread t2 = new Threadexample();
    t1.start();
    t2.start();
    try {
        sleep(200);
    } catch (InterruptedException e) { }
    if (t1.isAlive())
        System.out.println("Thread 0 is alive");
    else
        System.out.println("Thread 0 is dead");
    if (t2.isAlive())
        System.out.println("Thread 1 is alive");
    else
        System.out.println("Thread 1 is dead");
}
}

```

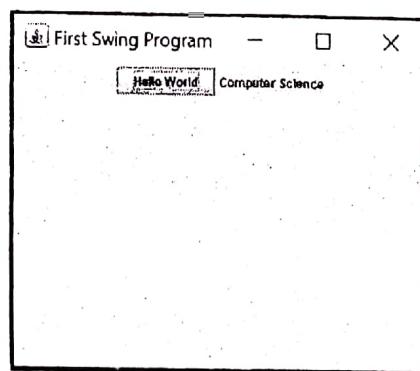
a) Briefly describe the following terms of java **Swing**: 4

i) Components ii) Layout Management

b) Write a java program using **FileInputStream** and **FileOutputStream** class to copy the content of one file located at c:\text1.txt to another file located at d:\text2.txt. 5

c) Create the following Graphical User Interface(GUI). Use **javax.swing** package to make the frame at location (100,150) which includes: 5

- button named “AUST”.
- label named “Computer Science”.



Date: 24/03/16

Ahsanullah University of Science and Technology

Department of Computer Science and Engineering
First Year, Second Semester, Final Examination (Fall 2015)

Course No: CSE 1205
Time: 3 Hours

Course Title: Object Oriented Programming
Full Marks: 70

[There are 7 (seven) questions carrying 14 marks each. Answer any 5 (five) questions.]
[Marks allotted are indicated in the right margin]

1.

- a) What are the three core concepts of Object Oriented Programming? Write down a java programme briefly explaining those three concepts. 8
- b) What are the differences between procedural and object oriented programming? 2
- c) Is java **call-by-value** or **call-by-reference**?- explain with an example. 3
- d) You write a java class named "MyProgramme.java". Write down two commands for compiling and running the programme in command line argument. 1

2.

- a) What do you understand by scope and lifetime of a variable? Discuss with suitable example with respect to block. 3
- b) Write down the output of the following sequence of code: 3

```
for (int i = 0; i < 8; i++) {  
    for (int j = 4 - (i % 4); j > 0; j--)  
        System.out.print(" ");  
    for (int j = 0; j < (i % 4) + 1; j++)  
        System.out.print("X");  
    System.out.println();  
}
```

- c) What is method overloading? Write down the rules of method overloading in java. 3
- d) Write a class named 'Book' which includes **book_name(String)**, **author_name (String)** and **price (double)**. This class should have the following constructors to initialize those instance variables:
 - A constructor without parameters
 - A constructor with three parameters
 - A constructor with object as parameter5

Create three objects of Book class using the above three constructors. Assign these values for those instance variables: ("The Complete Reference", "Herbert Schildt", 190). Also print the following information using **void displayInfo()** method:

"The book X of author Y costs Z"; where X,Y,Z will be replaced by those variables.

3.

- a) Briefly explain the public, private and protected access specifiers of java. 3
- b) Can a constructor have **private** access specifier? Explain with a suitable example. 5
- c) class A
{
 static int first=500;
 static String second;
 static
 {
 System.out.println("100");
 }
}

public class TestStatic
{
 static
 {
 System.out.println("300");
 }
 public static void main(String[] args)

```

        first = 100;
    }
    static
    {
        System.out.println("200");
        second = "SECOND";
    }
}

```

```

        System.out.println("400");
        System.out.println(A.first);
        System.out.println(A.second);
    }
}

```

What will be the output of above programme?

- d) "Class is logical construct and an object has physical reality" – explain the statement with example. 3

4.

- a) Consider the following piece of code:

3

```

Employee E1 = new Employee("Karim", 5001);
Employee E2 = new Employee(E1);

```

- b) Why should you use anonymous class? - explain with an example. 3
- c) Stack is a data structure known as Last In First Out (LIFO). You need to write down MyStack class which implements the interface IntStack. There will be two methods push() and pop() method in the interface. push() method will store an integer into stack and pop() method will retrieve an integer from the stack. 7

MyStack class has the following properties:

- An array of integers that can hold up to 25 integers.
- A stack pointer that will hold the current index of the stack
- A constructor to initialize the stack
- isEmpty method which will return true if the stack is empty otherwise return false
- isFull method which will return true if the stack is full otherwise return false

Now design your MyStack class implementing the IntStack interface.

- d) Write code snippet for infinite loop using while? 1

5.

- a) "A superclass variable can refer a subclass object", do you agree? Justify your answer. 3
- b) What are the difference between a class and an interface? 3
- c) What would be the output of the following statements? 3

```

class A {
    A() { System.out.println("Inside A"); }
}
class B extends A {
    B() { System.out.println("Inside B"); }
}
class C extends A {
    C() { System.out.println("Inside C"); }
}
class Example {
    public static void main(String[] args) {
        C obj = new C();
    }
}

```

- d) Why should you use the keyword super in your java program? Explain with example. 3
- e) Write down the conditions of return type in method overriding. 2

6

- a) Define a class named **classA** in a package **pA**. This class will contain two public int data member **a** and **b** and an abstract method **compute()**. Now define three classes named **classB**, **classC** and **classD** in packages **pB**, **pC** and **pD**. Each of the class is a subclass of **classA** and will implement the **compute()** method. In **classB**, it will compute the sum of **a** and **b**, in **classC**, compute the product of **a** and **b** and in **classD**, compute the result of subtraction. Now write a Java program that will generate call to the **compute()** method of each class. 6
- b) What are the three uses of **final** keyword? – explain briefly with suitable example. 3
- c) What is diamond problem in inheritance? How multiple inheritance is achieved in Java? 2
- d) What are the difference between **throw** and **throws**? 3

7.

- a) Write down the uses of the following methods: 5
(i)**compareTo**, (ii)**charAt**, (iii)**trim**, (iv)**subString** and (v)**lastIndexOf**
- b) Write a Java program that will take two integer numbers as input from the keyboard. Your program should determine whether the first number is a multiple of the second number. 6

Your program should provide checking for the following cases:

- i. If any of the two numbers is negative.
- ii. If the first number is smaller than the second number.
- iii. If the second number is 0.

You should define appropriate exception class (user defined) for each of the cases and throw an instance of the correct exception when any of the condition arises.

- c) Write short notes on Java Heap and Stack memory. 3

Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

First Year, Second Semester, Final Examination (Spring 2015)

Course No: CSE 1205

Course Title: Object Oriented Programming

Time: 3 Hours

Full Marks: 70

[There are 7 (seven) questions carrying 14 marks each. Answer any 5 (five) questions.]
[Marks allotted are indicated in the right margin]

1. a) What are the main three principles of Object Oriented Programming (OOP)? Explain them with the help of brief and proper example using Java language. 7
- b) Is java compiled language or interpreted language, or both? Give justification to your answer. 3
- c) Class Test{
 public static void main (String arg[]){
 System.out.println("My Favourite Language is Java");
 }
}

The above code is an example of java Class. Why do we need the main method? Why main method is defined as public and static? What does the parameter of the main method actually stand for?

2. a) You want to compile and run the above code 1(c) from the command line. Write down the commands for compiling and running the code. 1
- b) Briefly explain the automatic type conversion and type promotion in java. 4
- byte b = 60;
b = b * 2; // Error
Explain the reason for the error and correctly rewrite the second line.
- c) Design a class named **Student** which includes **firstName (String)** and **lastName(String)** as private instance variables. The default value of those variables will be your first name and last name. The class should have following constructors to initialize those instance variables.
 - A constructor without parameter
 - A constructor with two parameter
 - A constructor with object as parameter

There should be another method named **void display()** which will print the full name.

Create three objects of **Student** class using three different constructors and print the full name.

- d) What will be the output of the following code snippet? 2
- ```
int x = 10;
int y = 20;
if((x < y) || (x=5)>10) {
 Sysetm.out.println(x);
}else{
 System.out.println(y);
}
```
- e) Give an example to create an infinite loop using **while** in java? 1
3. a) Write a program that creates a multidimensional array of integers called **MyTwoD** like below. Use for loop to assign the following values and display each element of the array. 4

|    |    |    |
|----|----|----|
| 3  |    |    |
| 10 | 7  |    |
| 9  | 14 | 12 |

- b) A queue is a linear data structure which is known as FIFO (First In First Out). It means that the first element added to the queue will be the first one to be removed. The operation of addition in queue is known as Enqueue and the removal of the element known as Dequeue. Your task is to implement the queue in java class called **MyQueue**. 7

- I. An array of integers that can hold up to 20 integers
- II. A queue pointer will hold the current index of the queue
- III. A constructor to initialize the queue
- IV. enqueue( int v ) method will store an integer into the queue
- V. dequeue() method will retrieve an integer from the queue
- VI. printMyQueue() will print the elements of the queue in the order of their addition

Both enqueue() and dequeue() must handle queue overflow and underflow errors respectively.

c) class A{ 3  
     void method(double x){  
         System.out.println("x = " + x);  
     }  
  }  
 class B extends A{  
     void method(int y ){  
         System.out.println("y = " + y);  
     }  
     void method ( String s ){  
         System.out.println("s = " + s);  
     }  
     Public static void main ( String arg[]){  
         A a1 = new A();  
         A a2 = new B();  
         a1.method(10);  
         a2.method(20);  
         a2.method(13.45);  
     }  
 }

What will be the output of the above program?

4. a) What will be the output of the following program? 3

```
class VarArgs2 {

 static void vaTest(String msg, int ... v) {

 System.out.print(msg + v.length + " Contents: ");

 for(int x : v)

 System.out.print(x + " ");

 System.out.println();

 }

 public static void main(String args[]) {

 {

 vaTest("One vararg: ", 10);

 vaTest("Three varargs: ", 1, 2, 3);

 vaTest("No varargs: ");

 }

 }

}
```

- }
- b) Create a super class named **Student** with the following characteristics: 7
- **name(String)**, **address (String)** and **phNo (String)** are private instance variables
  - A constructor with three parameters that initializes instance variables.
  - An abstract method **print()** to display those instance variable
  - An abstract method **calculateScore ()** to calculate the score.

**Undergraduate** class with the following characteristics:

- Inherits **Student** class
- An array of 3 elements holding the three exams' score of a particular subject
- An instance variable named **totalExam** holding the number of exam that will be initialized with 3
- A constructor with its parameters that initializes instance variables. You should use **super()** to call the super-class constructor. Besides, your constructor should also have three variables for the exam's score. You should initialize the array with these three values.
- **calcluateScore()** method will return the maximum number from this array.
- Override **print()** method to display all the instance variable

**Graduate** class with the following characteristics:

- Inherits **Student** class
- An array of 4 elements holding the four exams' score of a particular subject
- An instance variable named **totalExam** holding the number of exam that will be initialized with 4
- A constructor with its parameters that initializes instance variables. You should use **super()** to call the superclass constructor. Besides, your constructor should also have four variables for the exam's score. You should initialize the array with these four values.
- **calcluateScore()** method will return the average of the two best exam.
- Override **print()** method to display all the instance variable

Design all the classes and also create a **Test** class. In this **Test** class; you should create one Undergraduate student and Graduate student and calculate the score and print the instance variables.

**Note: You should consider all the numbers are double.**

- c) Write down the conditions of the method overriding. 2
- d) Why a package is used? How can you use the codes in different packages? 2
5. a) What do you understand by **Dynamic Method Dispatch** in java? How does java achieve run-time polymorphism? Explain with an example. 4
- b) Explain **final**, **finally** and **finalize** keyword with an example. 3
- c) Check if the code will compile successfully. If not, correct the code. 3

```
interface NewShape
{
 void draw();
 int radius;
}
class NewCircle extends NewShape{
 public void draw(){
 radius=12;
 System.out.println("Radius is : "+radius);
 }
}
```

```

}
Class InterfaceTest{
 public static void main(String arg[]){
 NewShape nc1 = new NewCircle();
 nc1.draw();
 }
}

```

- d) Suppose, there is a class called **Account** having an instance variable **balance(double)** and a method **boolean withdrawAmount(double amount)** that read just the balance after withdrawing the amount. Update this method in java using the concept of exception handling. If the withdrawing amount is greater than the balance of the account, throw a user defined exception called "**MyWithdrawException**" to notify that "Withdraw operation fail. You do not have sufficient balance in your account".

Design the **Account** class and also the user defined exception class.

- 6 a) What are the differences between static method binding and dynamic method binding? 3
- b) If the main thread finishes earlier than the child thread then the program will misbehave. How can you force that the main thread will be finished last in java? Explain with an example. 2
- c) Write down the objectives of the three important methods of inter thread communication in java. 2
- d) What do you understand by thread synchronization in java? 5  
You need to design java synchronization for the following task with the help of synchronized block. There will be five threads and the task of each of the threads is to print number from 1 to 500. But no number must be printed more than once by the threads.

e) class MyThread extends Thread 2

```

{
 public static void main(String [] args)
 {
 MyThread t = new MyThread();
 t.start();
 System.out.print("one. ");
 t.start();
 System.out.print("two. ");
 }
 public void run()
 {
 System.out.print("Thread ");
 }
}

```

Why is there run time exception? Find out the problem.

7. a) "Java String is immutable" – explain with an example. 2  
b) What is the purpose of these following methods and give an example with them. 5

- i. **setPriority()**
- ii. **charAt()**
- iii. **trim()**
- iv. **join()**
- v. **equals()**

- c) What will be the output when you attempt to compile and run this code? 2

```

public class ExceptionTest
{
 public static void main(String [] args)
 {
 try
 {
 }
}

```

```
{
 badMethod();
 System.out.print("A");
}
catch (RuntimeException ex)
{
 System.out.print("B");
}
catch (Exception ex1)
{
 System.out.print("C");
}
finally
{
 System.out.print("D");
}
System.out.print("E");
}
public static void badMethod()
{
 throw new RuntimeException();
}
```

- d) What is String pool in java? How java uses string pool? 2
- e) Is java call by value or call by reference? – justify your answer with an example. 3

Date: 18.04.15

AHSANULLAH UNIVERSITY OF SCIENCE & TECHNOLOGY

Dept. of Computer Science and Engineering

Year: 1<sup>st</sup> Semester: 2<sup>nd</sup> Final Examination (Fall 2014)

Course No: CSE 1205  
Full Marks: 70

Course Title: Object Oriented Programming  
Time: 3 Hours

[There are Seven (7) Questions. Answer any Five (5) Questions.]  
[Marks allotted are indicated in the margin.]

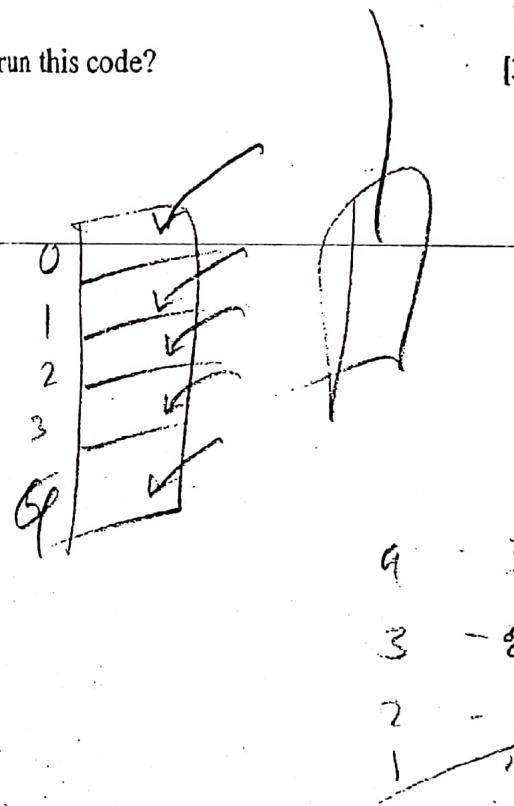
1. a) Briefly describe the features of an object-oriented programming. [4]
- b) What is the use of **return** statement? [2]
- c) Write short note on **Byte Code**. [3]
- d) Write a program that creates a 2-D array of integers called **TowDInt** like below. Use **for loop** for memory allocation. Then assign each array element with the following values. After that, use **enhanced for loop** to add all the elements of the 2-D array and show the summation as output. [5]
- |    |    |    |
|----|----|----|
| 15 | 25 | 35 |
| 10 | 20 |    |
| 5  |    |    |
- WheVet*
- Output: 110
2. a) Suppose, you are working for an online retail store that sells clothing. Your manager has asked you to modify the Customer class so that it assigns a new unique Customer ID number when a customer object is created. The current implementation constructs a new Customer object using the ID number supplied as a parameter. The Customer class has three variables named idNumber, name, mailingAddress; where all are private. [5]
- b) Explain with examples, the following terms in the context of java programming language. [6]
- final** keyword
  - finally** block
  - finalize** method
- c) What is the purpose of garbage collection in java? How to make an object for garbage collection? [2+1]
3. a) Create a class called queue with the following properties that implements a queue which can be used to store and retrieve numbers. Where retrieval is done in First-IN-First-Out (FIFO) order. [5]

- An array of numbers that can hold up to 50 numbers.
- A queue pointer that will hold the current index of the queue.
- A constructor to initialize the queue.
- Push() method to store a number into the queue.
- Pop() method to retrieve a number from the queue.

b) What will happen when you attempt to compile and run this code? [3]

```
abstract class Base{
 abstract public void myfunc();
 public void another(){
 System.out.println("Another method");
 }
}

public class Abs extends Base{
 public static void main(String args[]){
 Abs a = new Abs();
 a.amethod();
 }
 public void myfunc(){
 System.out.println("My Func");
 }
 public void amethod(){
 myfunc();
 }
}
```



c) What is the difference between function overriding and overloading? Explain clearly with simple code examples. [4]

d) Write down the uses of this keyword. [2]

4. a) Define the following classes. [6]

- Employee** class with following characteristics:
  - **ID**(integer), **name**(string) are private instance variables.
  - A constructor with two parameters that initializes member variables.
  - **print()** method displays all member variables.
  - Assume this class is stored in the package P1. Write also necessary code for this package information.
- Teacher** class with following characteristics:
  - Inherits Employee class.
  - **subject**(string), **name**(string) are private instance variables.
  - Overide **print()** method to display all instance variables by using **super** keyword.
  - Assume this class is stored in the package P2. Write also necessary code for this package information.

iii. Example class with following characteristics:

- Contains *main()* method.
- Assume this class is stored in the package P3. Write also necessary code for this package information.
- This class can evoke *print()* method of Teacher class.

b) Identify the errors in the following code segment with explanation. [3]

```
class outer{
 public int x=20;
 public int z=30;

 public void test(){
 inner ob= new inner();
 displayInner()

 }

 Public void displayOuter(){
 System.out.println("This is outer class");
 System.out.println("The value of y"+y);
 }
}

class inner{
 public int y=50;
 void displayInner(){
 System.out.println("This is inner class");
 displayOuter();
 }
}

class MainClass{
 public static void main(String args[]){
 outer ob=new outer();
 ob. displayOuter();
 ob. test();
 }
}
```

c) "A super class variable can reference a subclass object"- explain with an example. [3]

d) Write short note on **super** keyword. [2]

5. a) Create Staff\_Member class with following characteristics: [7]

- *name* (string), *address* (string) and *ph\_no*(string) are private instance variables.
- A constructor with three parameters that initializes instance variables.
- A *print()* method that displays those instance variables.
- A abstract method *double inc\_salary()* which increases salary by 1000.

Create Employee class with following characteristics:

- Inherits Staff\_Member class.
- salary*(double) is a public instance variable.
- A constructor with its parameters that initializes instance variables. You should use *super* to call the constructor of its super class.
- Overide *print()* method to display all instance variables by using *super* keyword.

Design all the classes. Create an object of Employee class and show the initialization of all the classes' instance variables. Then call *print()* method to display all the classes instance variables. Also call *inc\_salary()* method to display increased salary.

6) Write down the differences between Interface and Abstract class. [4]

✓) Identify the errors in the following code segment with explanation.

[3]

```
public class A {
 int x;
 void print(){
 System.out.println(x);
 }
}
public class B extends A {
 int y;
 void display(){
 System.out.println(y);
 }
 B(int a, int b){
 x=a;
 y=b;
 }
}
```

```
public class Example{
 public static void main(String[] args)
 {
 A ob=new A();
 B ob1=new B();
 ob.y=10;
 ob1=ob;
 ob1.display();
 }
}
```

✓ a) Consider a class **Employee** that contains array of String named “*name*”, “*address*” and an array of integer “*salary*”. [Same index contains the information of same employee] [6]

Now there is a interface named **IntSearch**; which contains a method *int search(String str)*. The “search” method in **Employee** class receives a name of an employee as parameter and return 1, if name is found in the “name” array and return 0, otherwise. In this class there is a *show()* method which displays –the salary and address of found employee, otherwise print – “No employee matched within the given array”.

Now create **EmployeeTest** class to demonstrate the capability of the problem.

b) Identify the errors in the following code segment with explanation.

[4]

```
public class SubClass {
 int sum=0;
 public final int NUMBER=10;

 SubClass(int x){
 NUMBER=x;
 }
 private void Add(){
 sum+=NUMBER;
 }
 private String toString(){
 return String.format("Sum %d", sum);
 }
}

public class MainClass {
```

```

public static void main(String[] args){
 SubClass obj=new SubClass(15);
 obj.Add();
 obj.NUMBER=15;
 System.out.printf("%s", obj);
}

```

c) With an example discuss how *Composition* works in java. [4]

7. a) Student class has two member variables *ID* and *CGPA*. Use parameterized constructor to initialize all the member variables. While initializing these variables, you have to consider the following restrictions:

-- ID must not be negative. Check ID whether negative or not and if found negative then throw an exception named "IDNegative".  
-- cgpa must be within 0 to 4; otherwise throw an exception named "CGPA Wrong".

- b) What will be the output when you attempt to compile and run this code? [3]

```

Class MainClass{
 public static void main(String[] args){
 int a[]={1};
 try{
 int x=10;
 int y=2;
 int z=x/y;
 a[13]=45;
 }
 catch(ArithmaticException ob){
 System.out.printf("%s", ob);
 }
 finally{
 System.out.printf("Finally block");
 }
 System.out.printf("After try catch block");
 }
}

```

- c) Draw the diagram of different states of *Thread*. [2]

- d) What are the differences between *Preemptive* and *Time Slicing Scheduling*? [2]

- e) Write down the uses of following methods:  
i. charAt() ii. getChars() iii. toCharArray() iv. indexOf() [2]