

Date: 15/03/201

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 example.

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

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

Sample Input: 845392

Sample output: 9

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

<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.

b) What do you understand by encapsulation in Java?

c) Define the following classes:

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.

Q) How can Java achieve multiple inheritance? Explain with example.

3

Q) What is the purpose of garbage collection in Java?

3

A) 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 is:

-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.

Q) What will be the output of following code?

4

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

Q) "Java String is called immutable" - explain with an example.

2

Q) Write down the differences between interface and abstract class.

3

Q) 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

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

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

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

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

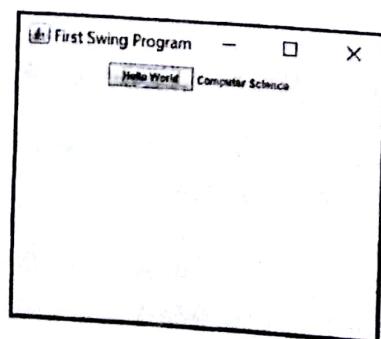
```
class Threadexample extends Thread {  
    public void run() {  
        System.out.println(Thread.currentThread().getName() + " is running...");  
        Thread t = Thread.currentThread();  
        if (Thread.currentThread().getName().equals("Thread-0")) {  
            System.out.println(t.getName()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

```

- X** a) Briefly describe the following terms of java **Swing**:
- Components
 - 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.
- c) Create the following Graphical User Interface(GUI). Use **javax.swing** package to make the frame at location (100,150) which includes:
- button named "AUST".
 - label named "Computer Science".



Ahsanullah University of Science and Technology

Date: 24/03/16

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

Course Title: Object Oriented Programming
Full Marks: 70

Course No: CSE 1205
Time: 3 Hours

[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 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 parameter

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)  
}
```

```

class A {
    static int first;
    static String second;
    static {
        System.out.println("1st");
        first = 100;
    }
    static {
        System.out.println("2nd");
        second = "SECOND";
    }
}

```

```

public class A {
    static {
        System.out.println("1st");
    }
    public static void main(String[] args) {
        System.out.println("400");
        System.out.println(A.first);
        System.out.println(A.second);
    }
}

```

a) Define a class and

What will be the output of above programme? 3
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);

```

What are the values of the expressions `E1.equals(E2)` and `E1 == E2`? Why? 3

- 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



BSC 12
@SC 12

Ahsanullah University of Science and Technology

Date: 11/10/15

Department of Computer Science and Engineering

First Year, Second Semester, Final Examination (Spring 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 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){  
    System.out.println(x);  
}else{  
    System.out.println(y);  
}
```

- e) Give an example to create an infinite loop using **while** in java? 1

- a) Write a program that creates a multidimensional array of integers called **MyTwoD** like below. 4
Use for loop to assign the following values and display each element of the array.

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**.

- 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{
    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?

- a) What will be the output of the following program?

```

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:
- 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.
- calculateScore() 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.
- calculateScore() 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 ncl = ne NewCircle();
            ncl.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. 5
 d) What do you understand by thread synchronization in java?
 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. 2

```

e) class MyThread extends Thread
{
    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?
- e) Is java call by value or call by reference? -justify your answer with an example.

Date: 18.04.15

AHSANULLAH UNIVERSITY OF SCIENCE & TECHNOLOGY

Dept. of Computer Science and Engineering

Year: 1st Semester: 2nd 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.]

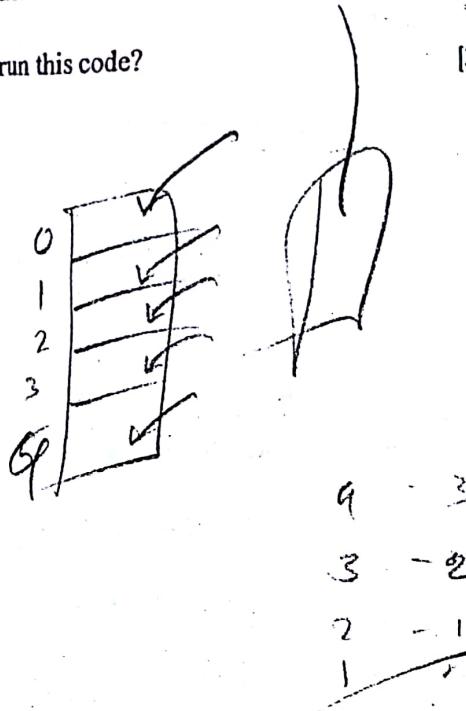
- ✓ a) Briefly describe the features of an object-oriented programming. [4]
- b) What is the use of *return* statement? [2]
- c) Write short note on *Byte Code*. [3]
- d) Write a program that creates a 2-D array of integers called *TowDInt* like below. Use *for loop* for memory allocation. Then assign 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 | | |
- Output: 110
- ✓ 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]
- ✓ 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.
 - Override *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.
- Override `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.

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

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

```
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();
```

```
}
```

6/ Q) 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]

- 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(ArithmeticException 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()

AHSANULLAH UNIVERSITY OF SCIENCE & TECHNOLOGY
Dept. of Computer Science and Engineering
Year: 1st Semester: 2nd Final Examination (Spring 2014)

Date: 20.8.14

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) What are *JVM* and *Byte Code*? How do they make Java platform independent? [4]
- b) What is encapsulation? Give an example. [3]
- c) With examples, write down the two uses of *break* statement. [2]
- d) Write a class *JavaStudent* which keeps track of how many students are present in the class. When a student enters into the class the total number of students increases. Again when a student exits the class the total number of students decreases. In main class declare 5 objects of *JavaStudent*, where each object creation increases number of students. Also print the total number of students in the class each time a student enters or departs. Invoke *departs()* method to decrease student number by 1. [5]
2. a) Write short note on *garbage collection* in java. [3]
- b) What will be the output of the following code segment? [3]
- ```
public class another {
 public int hour=1;
 public int mintue;
 private int second=3;

 public void settime(int hour,int mintue,int second){
 this.hour=hour; 3 5 9
 mintue=mintue; 6 5 9
 this.second=second;
 }

 public String gettime(){
 return String.format("%02d : %02d : %02d",hour,mintue,second);
 }
}
```

```
public class ThisUsed {
 public static void main(String args[]){
 another obj= new another();
 obj.settime(3, 5, 9);
 obj.hour=6;
 System.out.println(obj.gettime());
 }
}
```

- c) Write short note on final keyword. [2]

- 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 [6]

|     |                                                                                                                                                                                                                                     |     |     |     |     |    |    |     |     |     |    |    |    |     |     |     |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|-----|----|----|-----|-----|-----|----|----|----|-----|-----|-----|
| (1) | <table border="1"> <tr> <td>5</td><td></td><td></td><td>0 0</td></tr> <tr> <td>10</td><td>20</td><td>0,2</td><td>1 0</td><td>1 1</td></tr> <tr> <td>15</td><td>25</td><td>35</td><td>2 0</td><td>2 1</td><td>2 2</td></tr> </table> | 5   |     |     | 0 0 | 10 | 20 | 0,2 | 1 0 | 1 1 | 15 | 25 | 35 | 2 0 | 2 1 | 2 2 |
| 5   |                                                                                                                                                                                                                                     |     | 0 0 |     |     |    |    |     |     |     |    |    |    |     |     |     |
| 10  | 20                                                                                                                                                                                                                                  | 0,2 | 1 0 | 1 1 |     |    |    |     |     |     |    |    |    |     |     |     |
| 15  | 25                                                                                                                                                                                                                                  | 35  | 2 0 | 2 1 | 2 2 |    |    |     |     |     |    |    |    |     |     |     |

Output: 110

(2)  
0  
10  
15  
20  
25

3. a) Define nested class. With an example to show the scope of inner class and outer class. [4]
- b) Create a class named Executive having the following instance variable [7]

- salary (private, double)
- bonus (double)

Create a class named Worker having the following instance variables

- salary (double)
- hour (int)
- salaryPerHour (int)

Create a class Salary having method name calculateSalary overloaded two times. The first version can calculate the salary of Executive and the second version can calculate salary of Worker. The Executive's and Worker's payment are calculated by the following formulas.

Executive's payment = salary + bonus

Worker's payment = salary + hour \* salaryPerHour

Executive and Worker classes have corresponding constructor. Both classes have a method name show() which prints the value of all the instance variables and salary of corresponding classes (you have to call the salary class's overloaded method to print Executive's and Worker's salary).

You need to create two objects for Executive and Worker. Use parameterized constructor to initialize instance variables for each object and call the show() method.

- c) Identify the errors in the following code segment with explanation.

[3]

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

 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(5);
 obj.Add();
 obj.NUMBER=15;
 System.out.printf("%s", obj);
 }
}
```

[3]

4. a) Identify the errors in the following code segment with explanation.

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

c) What will be the output of  
Preemptive Scheduling?  
Preemptive TEST extends Test  
public void run()  
public void run()

- b) With an example discuss how *Composition* works in java. [4]
- c) Create Account class with the following characteristics:  
-- *name* (string), *account number*(string) are private instance variables.  
-- A constructor with two parameters that initializes instance variables.  
-- A *print()* method that displays those instance variables.  
--A abstract method *double withdraw(double amount)* which returns balance with the following formula:  
$$\text{balance} = \underline{\text{balance}} - \underline{\text{amount}}$$

Create Saving\_Account class with the following characteristics:

- Inherits Account class.
- *balance*(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 *Saving Account* class and show the initialization of all the classes' instance variables. Then call *print()* method to display all the classes instance variables. Also call *withdraw()* method to display balance after withdrawal.

5. a) Consider a class *Billing* that contains array of integer named *bill*. This class should have a [6] constructor to initialize the instance variable.

Now there is a interface named *BillSearch* which contains a method *int search(int b)*. The *search* method in *Billing* receives bill as parameter and find out the number of bills in *bill* array that is greater than equal to the bill which is received as parameter.

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

- b) What does *package* mean in java? What benefits *package* provides to java? [3]
- c) Write down the benefits of *Multi-Threading* in java. [2]
- d) What are the differences between *Preemptive* and *Time Slicing Scheduling*? [3]

6. a) Write a java program to throw an exception (user defined) for an employee details- [6]

- If an employee name's length is large than 20, a name exception must be thrown.
- If an employee age is greater than 50, an age exception must be thrown.

You need to handle another built in exception called "InputMismatchException" for age variable. The exception is thrown when your input variable type is not matched (Say, you variable is integer, but you have inputted String type).

The exception is in *java.util* package.

- b) What are the differences between *checked* and *unchecked exception*? [3]

- c) What will be the output of the following code segment? [Assume your system follows Preemptive Scheduling] [3]

```
public class TEST extends Thread {
 public void run(){
 System.out.println("Thread Name:"+currentThread().getName());
 System.out.println("Thread Priority:"+currentThread().getPriority());
 }
 public static void main(String[] args) {
 TEST t1= new TEST();
 TEST t2= new TEST();

 System.out.println("Name of Current Thread:"+
 currentThread().getName());

 t1.setPriority(1);
 t2.setPriority(10);
 t1.start();
 t2.start();

 t2.setName("THREAD_NEW");
 System.out.println("after changing Name of t2:"+ t2.getName());
 }
}
```

- d) Write a program to illustrate sub class exception precedence over base class. [2]
7. a) Write down the several ways to create *StringBuffers* in java. [2]
- b) Write down the prototypes and uses of following methods: [6]  
i) isAlive() ii) join() iii) setBounds() iv) setEnabled v) setCharAT() vi) toCharArray()
- c) Create the following GUI. Use java.awt package to make the form which includes two radio buttons and one check box, as displayed below. [6]

