



**Department of Computer and Information Sciences**

**52 141 Programming Foundations**

# **SAMPLE EXAM PAPER**

**(2 hours)**

**Question 1 is worth 3 marks. All other questions are worth 1 mark.**

**Indicate your answers on this examination paper.**

**For each question only one answer must be marked as correct.**

**A booklet is provided for rough working. Rough working is not marked. Your details should be completed on the front cover of the booklet and your registration number should be written at the bottom of this question paper. This question paper (containing your answers) must be attached inside the rough working booklet before submission.**

**THIS EXAM PAPER IS NOT OPEN BOOK –  
YOU CANNOT USE YOUR NOTES OR ANY  
TEXTBOOKS IN THE EXAM.**

1. For each of the following statements regarding object-oriented programming using Java, indicate whether you think it is true or false by circling the appropriate answer.

- a) Many similar objects can be created from a single class.  
**True / False**
- b) A method can only have one return type, or void.  
**True / False**
- c) The state of an object is defined by a combination of its setter and getter methods.  
**True / False**
- d) The programmer must provide a constructor for each new class that is defined.  
**True / False**
- e) The scope of a variable defines the section of source code from where the variable can be accessed.  
**True / False**
- f) Constructors are called automatically when a variable is declared.  
**True / False**
- g) An **ArrayList** implements the **Collection** interface.  
**True / False**
- h) Methods can only call other methods of the same class if those other methods are declared as **private**.  
**True / False**
- i) The body of a **while** loop may be executed zero, one or many times.  
**True / False**
- j) A class can have more than one direct superclass.  
**True / False**
- k) The static type of a variable **v** is the type of the object that is currently referenced by **v**.  
**True / False**
- l) Testing is the attempt to pinpoint and fix the source of an error.  
**True / False**

2. A Java class representing a bank account maintains the name as a **String** and the balance as a **double**. Which of the following is **NOT** a valid constructor for the class:

- ```
public Account(String n)
{
    name = n;
    balance = 0;
}
```
- ```
public Account(String n, double b)
{
    name = n;
    balance = b;
}
```
- ```
public Account()
{
    name = "";
    balance = 0.0;
}
```
- ```
public void Account(String n, double b)
{
    name = n;
    balance = b;
}
```
- ```
public Account(double b)
{
    name = "Bob";
    balance = b;
}
```
- ```
public Account(int b, String n)
{
    name = n;
    balance = b;
}
```

3. Which of the following statements regarding programming in Java are **true**:

- i) To use an object in a Java program, we first declare a variable, and then make it reference an object of appropriate type, and then send messages to it
- ii) The **double** data type has a larger range of values than the **int** data type
- iii) Sending a message to the '**null**' object causes a syntax error

- All of i), ii) and iii)
- Only i) and ii)
- Only i) and iii)
- Only ii) and iii)
- Only i)
- Only ii)
- Only iii)
- None of them

4. How many of the following 4 method declarations are legal Java:

```
private int sum(int x, int y) {  
    return x + y;  
}
```

```
public void difference(int x , int y) {  
    return x - y;  
}
```

```
public double multiply(double m, double n) {  
    m * n;  
}
```

```
public void divide(int x, int y) {  
    int d = x / y;  
}
```

- 0
- 1
- 2
- 3
- 4

5. Given the following **Bank** class definition, and assuming the **Bank** variables **bank1** and **bank2** are properly declared and **Bank** objects properly constructed, which of the following is **NOT** a valid statement in a class that uses the **Bank** class:

```
public class Bank
{
    private double amount;
    public void deposit(double money)
    {
        amount = amount + money;
    }
    public double getBalance()
    {
        return amount;
    }
}
```

- `bank1 = bank2;`
- `bank1.deposit( bank2.getBalance() );`
- `bank1.deposit(200.50);`
- `bank1.amount = bank2.amount;`
- `double d = bank1.getBalance()*2;`

6. What would be output by the following code fragment:

```
int age = 10;
String output = "Age is " + age;
output = output + 1;
System.out.println(output);
```

- Age is 10
- Age is 11
- Age is 101
- 11
- 101
- A syntax error

7. A Java class representing a book maintains the ISBN number as a **String** field called **isbn**. Which of the following method definitions correctly returns the **String** referenced by this field:

- ```
public void getIsbn()
{
    return isbn;
}
```
- ```
public getIsbn
{
    return isbn;
}
```
- ```
public getIsbn(String s)
{
    isbn = s;
}
```
- ```
public String getIsbn(String s)
{
    String isbn;
    return isbn = s;
}
```
- ```
public String getIsbn()
{
    return isbn;
}
```
- ```
public isbn getIsbn()
{
    return String;
}
```

8. Given the following method definition:

```
public int someMethod(int a)
{
    if (a > 0) {
        return a;
    }
    else if (a < 0) {
        a = -a;
        return a;
    }
    else {
        a = 0;
        return 0;
    }
}
```

and the method is used as follows in another class:

```
int x = -4;
int y = someMethod(x);
```

What are the values of the variables **x** and **y** after executing the statements above?

- x is 0, y is 0
- x is 4, y is 4
- x is -4, y is 4
- x is 4, y is -4
- x is -4, y is -4
- x is -4, y is 0

9. Which of the following are primitive data types in Java:

- i) **Collection**
- ii) **String**
- iii) **double**
- iv) **ArrayList**
- v) **Integer**
- vi) **boolean**

- All of them
- ii, iii, v and vi only
- iii, v and vi only
- ii, v and vi only
- iii and vi only
- None of them

10. What are the values of **a** and **b** following execution of the following code fragment:

```
int a = 3;
int b = 9;
while ( b >= 3 ) {
    a = a + b;
    b = b - 3;
}
```

- a is 3, b is 9
- a is 12, b is 0
- a is 18, b is 0
- a is 18, b is 3
- a is 21, b is 3
- a is 21, b is 0

11. What are the values of **b** and **c** after executing the following code fragment below:

```
boolean b = true;
int c = 0;
for ( int a = 0; a < 4; a++ ) {
    if ( a / 2 == 0 ) {
        c = c + 1;
        b = false;
    }
    else {
        c = c - 1;
        b = true;
    }
}
```

- b is true, c is 0
- b is false, c is 0
- b is true, c is -2
- b is false, c is -2
- b is true, c is -4
- b is false, c is 4

12. What is the value of **total** at the end of the loop:

```
int total = 0;
for ( int i = 0 ; i < 3; i++ ) {
    for ( int j = 3; j < 5; j++ ) {
        total = total + j;
    }
}
```

- 0
- 3
- 7
- 21
- 24
- 36

13. Consider the statements below.

- (i) A debugger can assist in locating the cause of syntax errors.
- (ii) An IDE such as Eclipse can assist in locating the cause of syntax errors
- (iii) When testing using assertions, a failing assertion indicates an error in the program.

Which of the above statements are true?

- only (iii)
- only (ii) and (iii)
- only (i)
- only (i) and (ii)
- only (ii)
- only (i) and (iii)

14. In an instance of a **HashSet** which one of the following is **false**?

- entering an element a second time has no effect
- each element is only stored at most once
- elements are maintained in order
- the method **size()** returns the current number of elements in the HashSet instance
- an iterator may return the elements from the HashSet in a different order from that in which they were entered

15. What is the value of **result** after executing the following fragment of code:

```
int a = 1;
char c = 'b';
int result = 0;

if ( ( a >= 1 ) && ( c > 'a' ) )
    result = result + 1;
else if ( ( a >= 1 ) || ( c > 'a' ) )
    result = result + 2;
else
    result = result + 3;
```

- 0
- 1
- 2
- 3
- 5
- 6

16. Assume that there is a class **Bank** which has an **amount** field of type **double** with methods **getAmount()** and **setAmount(double d)**. Consider the code fragment below:

```
Bank b1 = new Bank();
b1.setAmount( 150.50 );
Bank b2 = new Bank();
b2.setAmount( 10.25 );
b1 = b2;
b2.setAmount( 0.50 );
```

Which one of the following is true after executing the code fragment above?

- b1** refers to a Bank object with amount 10.25 and **b2** refers to a Bank object with amount 0.50
- b1** refers to a Bank object with amount 0.50 and **b2** refers to a Bank object with amount 10.25
- b1** and **b2** refer to different Bank objects both of which have amount 0.50
- b1** and **b2** refer to different Bank objects both of which have amount 10.25
- b1** and **b2** both refer to the same Bank object which has amount 0.50
- b1** and **b2** both refer to the same Bank object which has amount 10.25

17. Consider the statement below where **Student** is the name of a class:

```
Student [ ] studCollection = new Student[10];
```

What is the value returned by **studCollection.length** ?

- 0
- 1
- 9
- 10
- 11
- null

18. What is the value of **total** after the following code is executed:

```
ArrayList<Integer> numbers = new ArrayList<Integer>();  
numbers.add(1);  
numbers.add(2);  
numbers.add(3);  
numbers.add(2);  
numbers.remove(2);  
int total = 0;  
for (int i : numbers) {  
    total = total + i;  
}
```

- 0
- 3
- 4
- 5
- 6
- 8

19. Below is the definition of a class that holds an **array** of data of type **char**. The number of elements in the array is initialised on construction. The field **count** maintains a count of the number of elements that are currently used in the array.

```
public class TestClass
{
    private char[] chardata;
    private int count;

    public TestClass(int size)
    {
        chardata = new char[size];
        count = 0;
    }
}
```

Which of the following correctly defines a method to add a data item into the next available space in the array, returning **true** if there was space to add it and **false** if the array was already full?

`public boolean addItem(char item)`  
{  
    if (count < chardata.length) {  
        chardata[count] = item;  
        return true;  
    }  
    else  
        return false;  
}

`public boolean addItem(char item)`  
{  
    chardata[count] = item;  
    count++;  
    if (count < chardata.length) {  
        return true;  
    }  
    else  
        return false;  
}

`public boolean addItem(char item)`  
{  
    if (count < chardata.length) {  
        chardata[count] = item;  
        count++;  
        return true;  
    }  
    else  
        return false;  
}

`public boolean addItem(char item)`  
{  
    if (count < chardata.length) {  
        count++;  
        chardata[count] = item;  
        return true;  
    }  
    else  
        return false;  
}

20. Given class definition below:

```
public class TestClass {  
    private String[] names = {"Homer", "Marge", "Bart", "Lisa", "Maggie"};  
}
```

Which of the following method definitions, when incorporated into **TestClass**, would correctly search the names **array** for a specified **String** and return the position in the array where the first occurrence of the **String** is located (where **0** denotes the first position, **1** denotes the second, and so on) or **-1** if the **String** was not present in the array?

- ```
public int findpos(String s)  
{  
    int pos = 0;  
    for (int i = 0; i < names.length; i++) {  
        if (s.equals(names[i]))  
            pos = i;  
        else  
            pos = -1;  
    }  
    return pos;  
}
```
- ```
public int findpos(String s)  
{  
    for (int i = 0; i < names.length; i++) {  
        if (s.equals(names[i]))  
            return i;  
    }  
    return -1;  
}
```
- ```
public int findpos(String s)  
{  
    for (int i = 0; i < names.length; i++) {  
        if (!s.equals(names[i]))  
            return -1;  
    }  
    return i;  
}
```
- ```
public int findpos(String s)  
{  
    for (int i = names.length-1; i >=0; i--) {  
        if (s.equals(names[i]))  
            return i;  
    }  
    return -1;  
}
```

21. Given class definition below:

```
public class TestClass{  
    private String[] faculties= {"Science", "Engineering", "Arts and Social Science",  
                                "Education", "Business"};  
}
```

Which of the following is the correct header for a method within **TestClass** that returns an **array** of Faculty names sorted according to alphabetical order.

- `public Iterator sorted() {  
 // code to do sorting and return sorted array  
}`
- `public String sorted() {  
 // code to do sorting and return sorted array  
}`
- `public String[] sorted(String[]){  
 // code to do sorting and return sorted array  
}`
- `public faculties[] sorted(){  
 // code to do sorting and return sorted array  
}`
- `public ArrayList<String> sorted()  
{  
 // code to do sorting and return sorted array  
}`
- `public String[] sorted()  
{  
 // code to do sorting and return sorted array  
}`

22. Given the following definition of **StockItem** and **StockController** classes:

```
public class StockItem {
    private int id;
    private String name;
    private int quantity;

    public StockItem(int id, String name, int q) {
        this.id = id;
        this.name = name;
        quantity = q;
    }

    public int getID() {
        return id;
    }

    public String toString() {
        return id + ": " + name + " stock level: " + quantity;
    }
}
```

```
import java.util.*;

public class StockController {
    private ArrayList<StockItem> stock;

    public StockController() {
        stock = new ArrayList<StockItem>();
    }

    public void addStock(StockItem item) {
        stock.add(item);
    }

    public void printProductDetails() {
        Iterator<StockItem> it = stock.iterator();
        while(it.hasNext()) {
            StockItem item = it.next();
            System.out.println(item.toString());
        }
    }
}
```

Below is part of a test program used to test the two classes. If it is executed, what will be printed as a consequence of the last line?

```
StockController shop = new StockController();
shop.addStock(new StockItem(1006, "Spam", 105));
shop.addStock(new StockItem(1037, "Beans", 162));
shop.addStock(new StockItem(2031, "Taramasalata", 17));
shop.printProductDetails();
```

- 1037: Beans stock level: 162  
2031: Taramasalata stock level: 17
- 1006: Spam  
1037: Beans  
2031: Taramasalata
- 1006: Spam stock level: 105  
1037: Beans stock level: 162  
2031: Taramasalata stock level: 17
- 1006: Spam stock level: 105  
1037: Beans stock level: 162  
null

23. Given the following class definition:

```
public class League
{
    private HashMap<String, String> table;

    public League() {
        table = new HashMap<String, String>();
        buildLeague();
    }

    public String getPosition(String s) {
        String pos = table.get(s);
        return pos;
    }

    private void buildLeague() {
        table.put("1", "Killie");
        table.put("4", "Celtic");
        table.put("5", "Rangers");
        table.put("2", "QoS");
        table.put("3", "Ayr");
    }
}
```

What would be returned by the message `getPosition("2")`

- "4 Celtic"
- "5 Rangers"
- "2 QoS"
- "Celtic"
- "Rangers"
- "QoS"

24. Below is the definition of two classes - **Count** and **Worker**.

<pre>public class Count {     private int num;      public Count(int n) {         num = n;     }      public void inc() {         num++;     }      public int getCount() {         return num;     } }</pre>	<pre>public class Worker {     private Count c1;     private Count c2;      public Worker(Count t1, Count t2) {         c1 = t1;         c2 = t2;     }      public void work(int i) {         if (i &gt; 0)             c1.inc();         else             c2.inc();     }      public int status() {         if (c1.getCount() &gt; c2.getCount())             return c1.getCount() - c2.getCount();         else             return c2.getCount() - c1.getCount();     } }</pre>
---	---

In some test program the code below is executed:

```
Count a = new Count(4);  
Count b = new Count(4);  
Worker w = new Worker(a, b);  
System.out.print(w.status() + " ");  
w.work(2);  
System.out.print(w.status() + " ");  
w.work(-1);  
System.out.println(w.status() + " ");
```

What would be the sequence of output from the fragment of test code above?

- 0 1 0
- 0 1 1
- 0 2 0
- 4 1 0
- 4 2 -1
- 4 2 0
- 4 4 2 -1

**Questions 25 and 26 refer to the two classes Account and RockyAccount below:**

```
public class Account
{
    private double amount;
    public String name;

    public Account()
    {
        // body of constructor
    }

    public void add(double b)
    {
        // body of method
    }

    public double get1()
    {
        // body of method
    }
}

public class RockyAccount extends Account
{
    public double get2()
    {
        // body of method
    }
}
```

25. Consider the statements below with reference to classes **Account** and **RockyAccount** as defined above.

- i) **Account** is a subclass of **RockyAccount**
- ii) Both **Account** and **RockyAccount** implement the interface **Account**
- iii) A **RockyAccount** object can be assigned to a variable of type **Account**

Which of the above statements are true?

- All of i), ii) and iii)
- Only i) and ii)
- Only i) and iii)
- Only ii) and iii)
- Only i)
- Only ii)
- Only iii)
- None of them

26. Consider the statements below with reference to classes **Account** and **RockyAccount** as defined earlier.

- i) Sending message **add(20.5)** to an instance of **RockyAccount** would be valid
- ii) Field **name** can be referred to directly within the method bodies defined in class **RockyAccount**
- iii) Sending the messages **get1()** and **get2()** to an instance of **Account** would be valid

Which of the above statements are true?

- None of them
- Only i)
- Only ii)
- Only iii)
- All of i), ii) and iii)
- Only i) and ii)
- Only i) and iii)
- Only ii) and iii)

27. Given the class definitions below:

<pre>public class Class1 {      protected String res;      public Class1() {         res = "";     }      public Class1(String s1) {         res = s1;     }      protected String result() {         return res;     } }</pre>	<pre>public class Class2 extends Class1 {      public Class2(String s2) {         super(s2);     }      protected String result() {         return res + res;     } }</pre>
	<pre>public class Class3 extends Class1 {      public Class3(String s3) {         super(s3);     }      protected String result() {         return super.result();     } }</pre>

what output would be produced from the following fragment of code?

```
ArrayList<Class1> allcs = new ArrayList<Class1>( );  
allcs.add( new Class1("a") );  
allcs.add( new Class2("b") );  
allcs.add( new Class3("c") );  
allcs.add( new Class3("d") );  
allcs.add( new Class2("e") );  
allcs.add( new Class1() );  
  
for( Class1 c : allcs )  
    System.out.print(c.result());  
System.out.println( );
```

- abcde
- abcdef
- abbcdde
- abbcddeef
- abbccddee
- aabbccddee

28. Consider the following class definition.

```
public class Poetry {
    public void readPoem() throws IOException {
        File inFile = new File("poem.txt");
        FileReader fileReader = new FileReader(inFile);
        BufferedReader buffReader = new BufferedReader(fileReader);
        String s = buffReader.readLine();
        while (s != null) {
            if (s.contains("mi")) {
                System.out.println(s);
                s = buffReader.readLine();
            }
            s = buffReader.readLine();
        }
        buffReader.close();
    }
}
```

What output will the execution of `readPoem()` produce if the file `poem.txt` contains the data shown below?

```
right inuff
ma language is disgraceful
...
ma landlady in carrington street tellt mi
thi lassie ah tried tay get aff way in 1969 tellt mi
sum wee smout thit thoat ah hudny read chomsky tellt mi
a calvinistic communist thit thoat ah wuz revisionist tellt mi
```

- right inuff  
ma language is disgraceful  
...  
ma landlady in carrington street tellt mi  
thi lassie ah tried tay get aff way in 1969 tellt mi  
sum wee smout thit thoat ah hudny read chomsky tellt mi  
a calvinistic communist thit thoat ah wuz revisionist tellt mi
- ma landlady in carrington street tellt mi  
thi lassie ah tried tay get aff way in 1969 tellt mi  
sum wee smout thit thoat ah hudny read chomsky tellt mi  
a calvinistic communist thit thoat ah wuz revisionist tellt mi
- thi lassie ah tried tay get aff way in 1969 tellt mi  
a calvinistic communist thit thoat ah wuz revisionist tellt mi
- ma landlady in carrington street tellt mi  
sum wee smout thit thoat ah hudny read chomsky tellt mi
- ma landlady in carrington street tellt mi
- a calvinistic communist thit thoat ah wuz revisionist tellt mi
- a null pointer exception