



TECHNICAL UNIVERISTRY OF MOMBASA

# Faculty of Engineering & Technology

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

UNIVERSITY EXAMINATION FOR DEGREE IN:  
BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY  
(BSIT12S J-FT)

**BIT 2203: ADVANCED PROGRAMMING**

END OF SEMESTER EXAMINATION  
SERIES: DECEMBER 2014  
TIME: 2 HOURS

**Instructions to Candidates:**

You should have the following for this examination

- *Answer Booklet*

This paper consists of **FIVE** questions. Attempt question **ONE (Compulsory)** and any other **TWO** questions  
Maximum marks for each part of a question are as shown

This paper consists of **THREE** printed pages

---

**Question One (Compulsory)**

- a) Describe any **THREE** programming paradigms **(6 marks)**
- b) Distinguish between the following:  
(i) Problem domain and solution domain  
(ii) Bottom up and top down approaches **(8 marks)**
- c) Develop a Java class called Book with the following members:  
Data members: Title, Author, Publisher

**Question Two**

- a) Explain the term “Modularity” (2 marks)
- b) Explain TWO ways of implementing modularity in Java (6 marks)
- c) Using a suitable example show how you can implement modularity with one of the ways in (2b) above. (12 marks)

**Question Three**

- a) Explain the term design pattern as used in programming. (2 marks)
- b) State the main categories of design patterns (3 marks)
- c) Briefly describe any FIVE types of design patterns (15 marks)

**Question Four**

- a) Explain the term “generic type” as used in programming. (2 marks)
- b) Explain any FOUR types of generic classes in java (8 marks)
- c) Develop a generic java class called students with the following features:  
Data Members: Name, ID, Contact each of different generic type  
Method Members: Setters and Constructor (10 marks)

**Question Five**

- a) Explain any FOUR approaches used in problem solving. (8 marks)
- b) Consider the class below:

```
Public class patient < T, S, U>
{
Private T name;
Private S id;
Private U Contact ;
Public (T, n, S, d, U, c)
{
this.name = u;
this.id = id;
this.contact = C;
}
Public T getName ( )
{
return Name;
}
Public S get id ( )
{
return id;
}
Public U get contact ( )
```

```
{  
return contact  
}
```

Write an executable program called Patient Test that creates three instances of patients (JANE, 1, NRB); (TED, 2 MSA); (ANN, 3, ELD)

The program should display their names in ascending order

**(12 marks)**