

**COS1512  
RCO1512  
SECOND PAPER**

May/June 2017

**INTRODUCTION TO PROGRAMMING II**

Duration 2 Hours

75 Marks

**EXAMINERS**

FIRST

SECOND

DR MA SCHOEMAN  
MS A THOMAS

---

**Closed book examination**

**This examination question paper remains the property of the University of South Africa and may not be removed from the examination venue**

[TURN OVER]

**COS1512  
RCO1512  
SECOND PAPER**

May/June 2017

**INTRODUCTION TO PROGRAMMING II**

Duration 2 Hours

75 Marks

**EXAMINERS**

FIRST

SECOND

DR MA SCHOEMAN

MS A THOMAS

---

**Closed book examination**

**This examination question paper remains the property of the University of South Africa and may not be removed from the examination venue**

This paper consists of 6 pages and 6 questions

Please make sure that you have all 6 pages with the 6 questions

**Instructions:**

- Answer all the questions
- Do all rough work in the answer book
- The mark for each question is given in brackets next to the question
- Please answer the questions in the correct order. If you want to do a question later, leave enough space
- Number your answers and label your rough work clearly.
- Marks are awarded for part of an answer, so do whatever you are able to in each question

**GOOD LUCK!**

[TURN OVER]

**QUESTION 1****[5]**

Find the errors in the following statements. Explain the reason for each error, and provide the correct statements to achieve the desired effect

```
char str[5] = "Stop",
if (str != "STOP")
    str = "STOP",
```

**QUESTION 2****[4]**

A recursive function is a function that calls itself until a certain condition is met. Write a recursive function `CountDown` that accepts an `int` value as a starting value and displays a countdown to zero without using loops. The starting value must be greater than 0, if this condition is true call the function by passing a value less than the previous value and if the condition is false the function should terminate. E.g. when the call

```
CountDown(3)
```

is executed, the output should be

```
3
2
1
0
```

**QUESTION 3****[7]**

3.1 The code fragment below either contains an error or a risky statement. Identify and explain the error or risky statement

```
1 float* q,
2 q = new float(3 456),
3 delete q,
4 *q = 4.67, (1)
```

3.2 The following code fragment has some errors. Read the questions below and give the correct code for the lines in error

```
1 int * p1,
2 int * p2,
3 int x = 5,
4 int y = 10,
5 *p1 = x,
6 *p2 = y,
7 cout << p1 << " " << p2 ;
8 p1 = p2,
9 cout << *p1 + *p2 + 10,
```

[TURN OVER]

- 3 2 1 In lines 5 and 6 p1 and p2 are supposed to point to the variables x and y respectively Give the correct code for lines 5 and 6 (1)
- 3 2 2 Line 7 is supposed to display the values of the variables to which p1 and p2 are pointing Give the correct coding for line 7 (1)
- 3 2 3 What is displayed by line 9 after the corrections of 3 2 1 and 3 2 2 have been made? (1)
- 3 3 Write statements to create a dynamic array for which the size of the array can be obtained during program execution (3)

**QUESTION 4****[34]**

Define a class called `Product` that represents a single product sold by an office supply store. It should contain the following member variables

```
long id,           // 5 digit product ID number
double price,     // wholesale price
long number;      // number of products in stock
```

The class should contain the following member functions

- A default constructor that initializes `id` to 0, `price` to 0.0 and `number` to 0
- A destructor that does not perform any action
- An accessor function for member variable `number`
- A void member function called `reset` that resets the member variables of `Product` to values specified by parameters
- A void member function called `increaseRetailPrice()` that increases the `Product`'s price by a markup percentage as specified by a float parameter
- An overloaded equality operator `==` (implemented as a `friend` function) Use the following prototype

```
bool operator==(const Product& product1, const Product& product2)
```

This function returns `true` if `product1` is identical to `product2` and `false` otherwise

- Overload the prefix decrement operator `--` (implemented as a `friend` function) which returns the current instance of the class `Product` after decrementing `number` by 1. Use the following prototype
- An overloaded operator `>>` (implemented as a `friend` function) that inputs a `Product`'s `id`, `price` and `number` member variables from any input stream Use the following prototype
- An overloaded operator `<<` (implemented as a `friend` function) that inserts the values of a `Product`'s `id`, `price` and `number` member variables into an output stream Use the following prototype

```
Product operator--(Product& P)
istream& operator>> (istream& ins, Product& P)
ostream& operator<< (ostream& outs, const Product& P)
```

[TURN OVER]

Attempt the solutions as follows

- 4 1 Create the header file that contains the specification of the class Product (8)  
 4 2 Create the implementation of the class Product including all the friend functions (15)  
 4 3 Complete the application program (main()) below by citing the number and writing down the missing statement(s). This program obtains the details for a product from the user and resets an object to the values obtained from the user. It then extracts all existing products one by one from a file Products.dat and displays the product number of each product. The program also checks whether these products corresponds to the product information obtained from the user. If this is the case, the retail price for the product is increased by 5%, and the number of units in stock is decreased by one. Be sure to use appropriate member functions in the required statements. Do not rewrite the program, rather cite the line number and give your answer (11)

```
#include <iostream>
#include <fstream>
#include <cstdlib>
    1                               //1.Include files needed

using namespace std;

int main()
{
    2                               //2.Declare input file
    3                               //3.Open the file Products.dat and check
                                   // that the file exists

    Product P1,P2;
    long ID, Nr;
    float Price;
    cout << "Enter product ID ",
    cin >> ID;
    cout << "Enter wholesale product price. ",
    cin >> Price;
    cout << "Enter number in stock of product " << "ID:",
    cin >> Nr;

    4                               //4.Reset object P1 to the input values
                                   // obtained from the user

    while ( 5 ) //5.Extract an existing product from
                                   // file Products.dat ( use product P2)
    {
        6                               //6.Display product number

        if ( 7 ) //7.Compare product P1 with product P2
        {
            8                               //8.Increase retail price of P2 by 5%

            9                               //9.Decrement P2
        }
    }
}
```

[TURN OVER]

```

        10 //10.Display P1 and P2 on the console
    }
}
    11 //11.Close file

return 0;
}

```

**QUESTION 5****[14]**

Consider the class specification (interface) for the class Marks below

```

class Marks
{
public
    Marks(),
    Marks (string name, string number, int asg1, int asg2, int asg3,
           double test),
    double calcMark() const,
private
    string stdtName;
    string stdNumber;
    int[3] assignments,
    double testMark,
}

```

- 5.1 Derive a class FinalMark from class Marks. This class has an additional member variable examMark that holds the examination mark. The class should override member function calcMark() to calculate a final mark that includes the three assignments, the test mark and the examination mark.  
**NB:** Provide only the class interface (5)
- 5.2 Implement the overloaded constructor for the class FinalMark by invoking the base class constructor (3)
- 5.3 Implement the member function calcMark() for the derived class FinalMark. The test mark contributes 20% to the final mark, the average of the three assignments contributes 10% and the examination mark contributes 70% to the final mark. Note that function calcMark() must return an integer result (4)
- 5.4 Consider the following instantiation  
FinalMark myMark,  
Write down a statement to invoke the version of calcMark() provided in class Marks for object myMark and display the result on the console (1)
- 5.5 Indicate whether the following statement is valid or invalid. Give a reason for your answer.  
If testmark was a protected member variable of Marks, the following statements embedded in a complete main program would be legal  
Marks PetersMarks,  
cin >> PetersMarks.testMark, (1)

[TURN OVER]

**QUESTION 6****[11]**

Many application programs use a data structure called a dictionary in which one can use a key value to retrieve its associated data value. For example, we might want to associate automobile part numbers with the names of the corresponding parts

Key	Value
100000	tire
100001	wheel
100002	distributor
100003	air filter

The following class interface presents an approach to implementing the above scenario

```
class Dictionary {
public:
    Dictionary(),
    void Add(int key, const string &value),
    string Find(int key) const;
private:
    vector<int> Keys,
    vector<string> Values;
};
```

The class Dictionary has the following operations

- Add() - adds a new key and value to the dictionary
- Find() - retrieves the corresponding value for that particular key, for example Find(100002) would return "distributor"

- 6 1 Provide an interface of the template class Dictionary<Tkey,TValue> In other words, re-design the Dictionary interface so that it may be used to create a Dictionary containing keys and values of any type. For instance the value could be of type double, whereas the key could be of type char. Note the key and value may be of different types hence we need two different template arguments to be supplied (5)
- 6 2 Implement the Find() function of the template class Dictionary<Tkey,TValue>. Assume the key is valid and exists within the dictionary (5)
- 6 3 Provide a declaration for a Dictionary that has keys of type string and values of type double (1)