1.	Name of Course/Module	Computer Programming II
2.	Course Code	TCP1241
3.	Status of Subject	Core for B. Sc Bioinformatics
4.	MQF Level/Stage	Bachelor Degree – MQF Level 6
5.	Version (state the date of the last Senate approval)	June 2012
6.	Requirement for Registration	TCP1231 Computer Programming I
7.	Name(s) of academic/teaching staff	Chong Siew Chin Roy Chang Teo Chuan Chin
8.	Semester and Year offered	Trimester 2 (Beta Level)

9.	Objective of the course/module in the programme :
	To initiate skills to write algorithms and programs and to introduce various programming paradigms, programming methodology and object oriented language in C++.

10.	Learning Outcomes :
	At the completion of the subject, students should be able to:

LO1:   Identify the procedural programming and object-oriented programming paradigms. (Cognitive, Level 1)
LO2:   Identify the components of object-oriented program. (Cognitive, Level 2)
LO3:   Demonstrate the implementation of object oriented programming concepts (Cognitive, Level 3)
LO4:   Develop program in a high-level programming language correctly and effectively. (Cognitive, Level 5) |

11.	Synopsis:
	The major areas of study include: Comparison of procedure oriented, structure oriented and object oriented programming paradigms, Top-down design, algorithm development, Fundamentals of object-oriented design, Classes, Attributes and Behaviour, C++ fundamentals, Data members and member functions, Dynamic memory allocation, Concept of inheritance and polymorphism, Advanced C++ concepts and applications.
	Bidang pengajian meliputi: Perbandingan antara pengaturcaraan bertatacara, berstruktur dan berobjek, Rekabentuk atas-bawah, Pembangunan algoritma, Rekabentuk berobjek asasi, Kelas, Atribut dan Tingkahlaku, C++ asasi, Ahli data dan Fungsi ahli, Peruntukan ingatan dinamik, Konsep-konsep warisan dan polimorfisma, Konsep dan aplikasi C++ Lanjutan.

12.	Mapping of Subject to Programme Outcomes :	
	Programme Outcomes	% of Contribution
	PO1: Apply soft skills in work and career related activities	50

	PO2: Demonstrate knowledge and understanding of fundamental concepts, principles and best practices	50

13.	Assessment Methods and Types :		
	Method and Type	Description/Details	Percentage
	Lab Test	Practical Test	20%
	Quiz	Written Quiz	20%
	Assignment	Report & Demo	20%
	Final Exam	Written Exam	40%

14.			
	Details of Subject		
	Topics	Mode of Delivery	
		Lecture	Lab
	1. Introduction Comparison of procedure oriented, structure oriented and object oriented programming paradigms. Top-down design. Algorithm development. Refining algorithms. Fundamental of object oriented design. Identifying classes. Attributes and behaviour. Features of object oriented programming.	2	2
	2. C++ Fundamentals Standard I/O streams. Function prototypes. C++ enhancements to C. Default function parameters. Inline functions. Overloaded functions. Reference variables comparison between pointers and references.	4	4
	3. Classes Creating new data type in C++. Class declaration. Members. Constructors and destructors. Access functions constant objects. Member objects. Static members. Friend classes. Arrays of class objects.	4	4
	4. Dynamic Memory Allocation Free store. New and delete operators. Class with pointer members. This pointer assignment. Initialization. Copy constructor. Passing and returning objects. Advanced free store techniques. Exception handling.	5	5
	5. Inheritance and Polymorphism Operator overloading. Handling related types in C++ - derived class. Conversion between base and derived classes. Virtual functions. Dynamic binding. Pure virtual functions. Protected members. Public and private base classes. New, delete operators overloading. Inheritance applications.	6	6
	6. Advanced C++ concepts an applications File handling. Templates. Container classes. Class library. Stack, queue and linked list applications. Simple database applications.	5	5
	Total	**26**	**26**

15.	Laboratory		
	- Simple C++ programs. - Classes and objects. - Polymorphism (function overloading and operator overloading) - Inheritance - File handling, templates and simple data structures.		

16.	Total Student Learning Time (SLT)	Face to Face (Hour)	Total Guided and Independent Learning
	Lecture	26	26
	Tutorials	-	-
	Laboratory/Practical	26	13
	Presentation	-	-
	Assignment	1	10
	Lab Test	2	4
	Final Exam	2	15
	Quizzes	6 times	6
	Sub Total	57	74
	Total SLT	131/40 = 3.275 => 3	

17.	Credit Value	3

18.	Reading Materials :	
	Textbook	Reference Materials
	1. Tony Gaddis, Judy Walters, Godfrey Muganda, "Starting out with C++: Early Objects", Pearson, 7th Ed, 2011.	1. Walter Savitch, "Problem Solving with C++", Addison Wesley, 8th Ed, 2011. 2. Y. Daniel Liang, "Introduction to Programming with C++", Pearson, 2nd Ed, 2009. 3. Paul Deitel, Harvey Deitel, "C++ How to Program", Pearson, 8th Ed, 2011.

19.	Appendix (to be compiled when submitting the complete syllabus for the programme) :
	     1. Mission and Vision of the University and Faculty
	     2. Mapping of Programme Objectives to Vision and Mission of Faculty and University
	     3. Mapping of Programme Outcome to Programme Objectives
	     4. Programme Objective and Outcomes (Measurement and Descriptions)