**Syllabus for International Mathematics Olympiad (IMO)**

The syllabus for Mathematics Olympiad (regional, national and international) is pre-degree college
mathematics.
The areas covered are

Number Theory - Arithmetic of integers
Divisibility( primes, gcd, lcm, Euclidean Algorithm)
Congruence (Fermat Little Theorem, Chinese Remainder Theorem, Wilson's Theorem , Euler's Toutient Theorem etc)
Number Theoretic functions ( Ceiling and floor functions)
p- adic Valuation
Diophantine Equation (Pell's equation)

Geometry - Points, lines and polygons
Trigonometry
Circumcentre, Incentre, Orthocentre, Centroid
Theorems of Ceva and Menalaus
Quadrilaterals
Circle Geometry
Other Theorems ( Nine-point circle, Stewart's theorem, Ptolemy's theorem, Simpson's line, Euler's line etc)
Coordinate geometry
Complex Geometry
Baricentric Geometry
Inversion Geometry
Projective Geometry

Algebra - Sequence and series ( A.P, G.P, Harmonic sequence, Fibonacci sequence)
Functional Equations
Polynomials (systems of equations, Viete's functions, factorizations etc)
Inequalities (AM-GM inequality, Cauchy- Swartz inequality, Chebychev' s inequality, Holder's inequality, Minkowski's inequality , Jensen's inequality, Schur's inequality, Muirhead'sinequality ,Trigonometric inequality, etc)

Combinatorics - Principle of Counting ( Permutation and combinations)
Principle of Mathematical Induction (PMI)
Inclusion- Exclusion Principle (Derangement)
Pigeon-hole Principle
Invariant Principle
Graph Theory

**<u>Note:</u>** The syllabus does not include Calculus and Statistics. The syllabus is in a sense spread over Class IX to Class XII levels, but the problems under each topic are of exceptionally high level in difficulty and sophistication. The difficulty level increases from RMO to INMO to IMO.

**Syllabus for International Chemistry Olympiad (IChO)**
**THEORETICAL**

**1. The atom**

**2. Chemical bonding**

**3. Chemical calculations**

**4. Periodic trends**

**5. Inorganic Chemistry**

1. Introduction
2. Groups 1 and 2
3. Groups 13 – 18 and Hydrogen
4. Transition elements
5. Lanthanides and actinides 3
6. Coordination chemistry including stereochemistry
7. Selected industrial processes

**6. Physical chemistry**

1. Gases
2. Thermodynamics

3. Equilibrium
4. Electrochemistry

**7. Chemical kinetics (Homogeneous reactions)**
1. Rate law
2. Reaction mechanisms

**8. Spectroscopy**
1. UV/visible
2. Infrared
3. x-Ray
4. NMR
5. Mass spectrometry

**9. Organic Chemistry**

**10. Polymers**

**11. Biochemistry**
1. Carbohydrates

**12. Analytical chemistry**

**EXPERIMENTAL**

**Inorganic Chemistry**

- Reactions of s-block elements with water, oxygen and halogens, their color in flame tests.
- Stoichiometry, reactions and properties of binary non-metal hydrides.
- Common reactions of carbon, nitrogen and sulfur oxides (CO, CO2, NO, NO2, N2O4, SO2, SO3).
- Common oxidation states of p-block elements, stoichiometry of common halides and oxoacids (HNO2,
HNO3, H2CO3, H3PO4, H3PO3, H2SO3, H2SO4, HOCl, HClO3, HClO4).
- Reaction of halogens with water.
- Common oxidation states of first row transition metals (Cr(III), Cr(VI), Mn(II), Mn(IV), Mn(VII), Fe(II),

Fe(III), Co(II), Ni(II), Cu(I), Cu(II), Ag(I), Zn(II), Hg(I), and Hg(II) )and the color of these ions.

- Dissolution of these metals and Al, amphoteric hydroxides (Al(OH)3, Cr(OH)3, Zn(OH)2).
- Permanganate, chromate, dichromate ions and their redox reactions.
- Iodometry (reaction of thiosulfate and iodine).
- Identification of Ag+, Ba2+, Fe3+, Cu2+, Cl–, CO3
2–, SO4
2– .

**Organic Chemistry**

- Common electrophiles and nucleophiles.
- Electrophilic addition: addition to double and triple bonds, regioselectivity (Markovnikoff's rule), stereochemistry.
- Electrophilic substitution: substitution on aromatic rings, influence of substituents on the reactivity and regioselectivity, electrophilic species.
- Elimination: E1 and E2 reactions at sp3 carbon centers, stereochemistry, acid-base catalysis, common leaving groups.
- Nucleophilic substitution: SN1 and SN2 reactions at sp3 carbon centers, stereochemistry.
- Nucleophilic addition: addition to carbon-carbon and carbon-hetero atom double and triple bonds, addition-elimination reactions, acid-base catalysis.
- Radical substitution: reaction of halogens and alkanes.
- Oxidations and reductions: switching between the different oxidation levels of common functional groups (alkyne – alkene – alkane – alkyl halide, alcohol – aldehyde, ketone – carboxylic acid derivatives, nitriles – carbonates).
- Cyclohexane conformations.
- Grignard reaction, Fehling and Tollens reaction.
- Simple polymers and their preparation (polystyrene, polyethylene, polyamides, polyesters).

**Biochemistry**

Amino acids and their classification in groups, isoelectric point, peptide bond, peptides and proteins.

Carbohydrates: open chain and cyclic form of glucose and fructose.

Lipids: general formulae of triacyl glycerides, saturated and unsaturated fatty acids.

## Syllabus for International Physics Olympiad (IPhO)

### General

A. The extensive use of the calculus (differentiation and integration) and the use of complex numbers or solving differential equations should not be required to solve the theoretical and practical problems.

B. Questions may contain concepts and phenomena not contained in the Syllabus but sufficient information must be given in the questions so that candidates without previous knowledge of these topics would not be at a disadvantage.

C. Sophisticated practical equipment likely to be unfamiliar to the candidates should not dominate a problem. If such devices are used then careful instructions must be given to the candidates.

D. The original texts of the problems have to be set in the SI units.

### A. THEORETICAL PART

### 1. MECHANICS

a] Foundation of kinematics of a point mass

* Note: vector description of the position of the point mass, velocity and acceleration as vectors

b] Newton's laws, inertial systems

* Note: problems may be set on changing mass

c] Closed and open systems, momentum and energy, work, power

d] Conservation of energy, conservation of linear momentum, impulse

e] Elastic forces, frictional forces, the law of gravitation, potential energy and work in a gravitational field

* Note: Hooke's law, coefficient of friction (F/R=const), frictional forces static and kinetic, choice of
zero of potential energy
f] Centripetal acceleration, Kepler's laws

**2] MECHANICS OF RIGID BODIES**
a] Statics, center of mass, torque
* Note: couples, conditions of equilibrium of bodies
b] Motion of rigid bodies, translation, rotation, angular velocity, angular acceleration, conservation of
angular momentum
* Note: conservation of angular momentum about fixed axis only
c] External and internal forces, equation of motion of a rigid body around the fixed axis, moment of inertia,
kinetic energy of a rotating body
* Note: parallel axes theorem (Steiner's theorem), additivity of the moment of inertia
d] Accelerated reference systems, inertial forces
* Note: knowledge of the Coriolis force formula is not required

**3] HYDROMECHANICS**
No specific questions will be set on this but students would be expected to know the elementary concepts
of pressure, buoyancy and the continuity law.

**4] THERMODYNAMICS AND MOLECULAR PHYSICS**
a] Internal energy, work and heat, first and second laws of thermodynamics
* Note: thermal equilibrium, quantities depending on state and quantities depending on process
b] Model of a perfect gas, pressure and molecular kinetic energy, Avogadro's number, equation of state of a
perfect gas, absolute temperature
* Note: also molecular approach to such simple phenomena in liquids and solids as boiling, melting etc.
c] Work done by an expanding gas limited to isothermal and adiabatic processes
* Note: proof of the equation of the adiabatic process is not required

d] The Carnot cycle, thermodynamic efficiency, reversible and irreversible processes, entropy (statistical
approach), Boltzmann factor
* Note: entropy as a path independent function, entropy changes and reversibility, quasistatic
processes

**5] OSCILLATIONS AND WAVES**
a] Harmonic oscillations, equation of harmonic oscillation
* Note: solution of the equation for harmonic motion, attenuation and resonance - qualitatively
b] Harmonic waves, propagation of waves, transverse and longitudinal waves, linear polarization, the
classical Doppler effect, sound waves
* Note: displacement in a progressive wave and understanding of graphical representation of the wave,
measurements of velocity of sound and light, Doppler effect in one dimension only, propagation of
waves in homogeneous and isotropic media, reflection and refraction, Fermat's principle
c] Superposition of harmonic waves, coherent waves, interference, beats, standing waves
* Note: realization that intensity of wave is proportional to the square of its amplitude. Fourier analysis
is not required but candidates should have some understanding that complex waves can be made
from addition of simple sinusoidal waves of different frequencies. Interference due to thin films and
other simple systems (final formulas are not required), superposition of waves from secondary
sources (diffraction)

**6] ELECTRIC CHARGE AND ELECTRIC FIELD**
a] Conservation of charge, Coulomb's law
b] Electric field, potential, Gauss' law
* Note: Gauss' law confined to simple symmetric systems like sphere,cylinder, plate etc., electric dipole

moment
c] Capacitors, capacitance, dielectric constant, energy density of electric field

**7] CURRENT AND MAGNETIC FIELD**

a] Current, resistance, internal resistance of source, Ohm's law, Kirchhoff's laws, work and power of direct
and alternating currents, Joule's law
* Note: simple cases of circuits containing non-ohmic devices with known V-I characteristics
b] Magnetic field (B) of a current, current in a magnetic field, Lorentz force
* Note: particles in a magnetic field, simple applications like cyclotron, magnetic dipole moment
c] Ampere's law
d] Law of electromagnetic induction, magnetic flux, Lenz's law, self-induction, inductance, permeability,
energy density of magnetic field
* Note: magnetic field of simple symmetric systems like straight wire, circular loop and long solenoid
e] Alternating current, resistors, inductors and capacitors in AC-circuits, voltage and current (parallel and
series) resonances
* Note: simple AC-circuits, time constants, final formulae for parameters of concrete resonance circuits
are not require

**8] ELECTROMAGNETIC WAVES**

a] Oscillatory circuit, frequency of oscillations, generation by feedback and resonance
b] Wave optics, diffraction from one and two slits, diffraction grating, resolving power of a grating, Bragg
reflection
c] Dispersion and diffraction spectra, line spectra of gases
d] Electromagnetic waves as transverse waves, polarization by reflection, polarizers
* Note: superposition of polarized waves
e] Resolving power of imaging systems
f] Black body, Stefan-Boltzmanns law

* Note: Planck's formula is not required

**9] QUANTUM PHYSICS**

a] Photoelectric effect, energy and impulse of the photon
* Note: Einstein's formula is required
b] De Broglie wavelength, Heisenberg's uncertainty principle

**10] RELATIVITY**

a] Principle of relativity, addition of velocities, relativistic Doppler effect
b] Relativistic equation of motion, momentum, energy, relation between energy and mass, conservation of
energy and momentum

**11] MATTER**

a] Simple applications of the Bragg equation
b] Energy levels of atoms and molecules (qualitatively), emission, absorption, spectrum of hydrogenlike
atoms
c] Energy levels of nuclei (qualitatively), alpha-, beta- and gamma-decays, absorption of radiation, halflife
and exponential decay, components of nuclei, mass defect, nuclear reactions.

**B. PRACTICAL PART**

The Theoretical Part of the Syllabus provides the basis for all the experimental problems. The
experimental problems given in the experimental contest should contain measurements.
Additional requirements:
1] Candidates must be aware that instruments affect measurements.
2] Knowledge of the most common experimental techniques for measuring physical quantities mentioned in
Part A.
3] Knowledge of commonly used simple laboratory instruments and devices such as calipers, thermometers,
simple volt-, ohm- and ammeters, potentiometers, diodes, transistors, simple optical devices and so on.

4] Ability to use, with the help of proper instruction, some sophisticated instruments and devices such as double-beam oscilloscope, counter, ratemeter, signal and function generators, analog-to-digital converter connected to a computer, amplifier, integrator, differentiator, power supply, universal (analog and digital) volt-, ohm- and ammeters.

5] Proper identification of error sources and estimation of their influence on the final result(s).

6] Absolute and relative errors, accuracy of measuring instruments, error of a single measurement, error of a series of measurements, error of a quantity given as a function of measured quantities.

7] Transformation of a dependence to the linear form by appropriate choice of variables and fitting a straight line to experimental points.

8] Proper use of the graph paper with different scales (for example polar and logarithmic papers).

9] Correct rounding off and expressing the final result(s) and error(s) with correct number of significant digits.

10] Standard knowledge of safety in laboratory work. (Nevertheless, if the experimental set-up contains any safety hazards the appropriate warnings should be included into the text of the problem.)

**Please note:**

□□The syllabus for National Standard Examination in Physics (NSEP) is broadly equivalent to the senior secondary level (Class XI and Class XII) of CBSE Physics. This is only a rough guideline, and there is no detailed syllabus given for NSEP.

□□The syllabus for Indian National Physics Olympiad (INPhO) is broadly similar to NSEP but the difficulty level of the questions will be higher. Questions and problems in National Olympiads are usually nonconventional and of high difficulty level, comparable to International Olympiads.

## Syllabus for International Olympiad on Biology (IBO)

### Syllabus for Theoretical Component

The IBO theoretical examination should concentrate on biological concepts applied to the majority of
organisms of the same group. It should not contain specific facts, exceptions or knowledge about local
organisms that require special or local experiences. The majority of questions should test students'
understanding, science
process skills and application of their biological knowledge. The questions testing only knowledge should be
as few as possible and they should not exceed 25 % of the total points.
After approval of the test questions by the International Jury the maximum obtainable points for correct
answers of each particular question have to be stated in the examination papers.
Questions concerning Principles of Scientific Reasoning and Principles of Biological Methods should be
included in the Theoretical test, which should cover the following 7 topics in the indicated proportions.
In the IBO tasks the names of organisms will be the national names (no description) together with the
scientific names (Latin) in brackets.
Any description instead of name is prohibited. The organizers should construct the questions so that the
name of the organism is not a key element for answering; otherwise they should use very well known
organisms (general representatives of a group) mentioned in the list for biosystematics.

### I Cell biology

### Structure and function of cells

- Enzymes

. Chemical structure: apoenzyme and coenzyme

. Model for enzyme action: enzyme binds with substrate

. Denaturation

. Nomenclature
- Nucleic Acids : DNA, RNA
- Other important compounds
. ADP and ATP
. NAD+ and NADH
. NADP+ and NADPH
* Organelles
nucleus - nuclear envelope
- (nucleohyaloplasm)
- chromosomes
- nucleoli
cytoplasm - cell membrane
- hyaloplasm
- mitochondria
- endoplasmatic reticulum
- ribosomes
- Golgi apparatus
- lysosomes
- vacuole membrane
- proplastides
- plastides .chloroplasts
. chromoplasts
. leucoplasts)
(e.g. amyloplasts)
Plant cells are surrounded with a cell wall
* Cell metabolism
- Breakdown of carbohydrates
. Anaerobic break down (anaerobic respiration) of glucose:
glycolysis
. Aerobic break down (aerobic respiration) of glucose: glycolysis
citric acid cycle
oxidative phosphorylation
- Dissimilation of fats and proteins
- Assimilation
. Photosynthesis
. Light reaction
. Dark reaction (Calvin cycle)

* Protein synthesis
- Transcription
- Translation
- Genetic code
* Transport through membranes
- Diffusion
- Osmosis, plasmolysis
- Active transport
* Mitosis and meiosis
- Cell cycle: interphase (replication) and mitosis (prophase - metaphase - anaphase - telophase)
- Chromatids, equatorial plate, haploid and diploid, genome, somatic and generative cells, gamete, crossing over
- Meiosis I and meiosis II.

**Microbiology**
* Prokaryotic cell organization
* Morphology
* Phototrophy and chemotrophy

**Biotechnology**
* Fermentation
* Genetic manipulation of organisms

**II Plant anatomy and physiology (15 %)**
(with emphasis on seed plants)
Structure and function of tissues and organs involved in:
* Photosynthesis, transpiration and gas exchange
- Leaf : structure; function stomata
* Transport of water, minerals and assimilates
- Root : structure (endodermis)
- Stem : structure (vascular bundles)
* Growth and development
- Apical meristem and cambium
- Germination
* Reproduction (ferns and mosses included)
- Asexual reproduction (clone forming)

- Sexual reproduction
. Structure of flowers
. Pollination
. Double fertilization
- Alternation of generation in seed plants, ferns and mosses

**III Animal anatomy and physiology (25 %)**
(with emphasis on vertebrates and especially man)
Structure and function of organs and tissues involved in
* Digestion and nutrition
- Digestive tract (including liver, gall bladder and pancreas)
- Mechanical and chemical breakdown of food
- Absorption
- Food components (water, minerals, vitamins, proteins, carbohydrates and fats)
* Respiration
- Breathing mechanism
- Gas exchange
- Respiratory organs
* Circulation
- Blood : blood plasma, red blood cells, white blood cells, blood platelets
- Blood circulation : arteries, capillaries, veins, heart
- Lymphatic system : tissue fluid, lymph
* Excretion
- Structure of the renal system
- Urine production
* Regulation (neural and hormonal)
- Nervous system : peripheral nervous system, central nervous system (spinal cord and brain), autonomic nervous system (sympathetic and parasympathetic), reflexes, sense organs (eyes and ears)
- Endocrine system : pituitary gland, thyroid gland, islets of Langerhans, adrenal medulla, adrenal cortex, ovaries and testes

* Reproduction and development
- Structure and function of male and female reproductive systems
- Ovulation and menstrual cycle
- Fertilization
- Formation of ectoderm, mesoderm, endoderm
- Embryonic membranes
* Immunity
- Antigens, antibodies

**IV Ethology (5 %)**

* Methodology of Ethology
* Innate and Learned Behavior
* Communication and Social Organization
* Foraging Behavior
* Defensive Behavior
* Mating systems and Parental care
* Biorhythms

**V Genetics and Evolution (20 %)**

* Variation : mutation and modification
* Mendelian inheritance
- Monohybrid cross
- Dihybrid cross
- Polyhybrid cross
* Multiple allelism, recombination, sex linkage
* Hardy-Weinberg principle
* Mechanism of evolution
- Mutation
- Natural selection
- Reproductive isolation
- Adaptation
- Fitness

**VI Ecology (10 %)**
* Individual Organisms
- Unitary and modular organisms
* Population
- Population structure
. dispersion, age, size and sex structure
- Population dynamics
. birth rate, death rate
. exponential and logistic growth, carrying capacity
- Population regulation
. metapopulation dynamics
* Biotic Communities
- Species reachness and diversity
- Niche, competition exclusion principle
- Interspecific Interactions
. competition, predation, symbiosis
- Community dynamics
. succession
- Terrestrial biomes

- Aquatic biomes
* Ecosystems
- Trophic structure
. food webs
- Trophic levels
. producers, consumers, decomposers
- Energy flow
- Productivity
. gross and net primary productivity
. energy transfer efficiencies
- Matter flux through ecosystems
- Global biogeochemical cycles
* Biosphere and man
- Human population growth
- Pollution
. threats of biodiversity

. in situ conservation
. ex situ conservation

**VII BIOSYSTEMATICS (5 %)**

- Structure and function, evolutionary and ecological relationships among typical organisms in the following groups. Knowledge of scientific terms will not be required for successful solution of the tasks. However, competitors should know what the named representatives of genera mentioned below look like.

Domain Archaea Methanobacterium,
Halobacterium,Thermoplasma,
Sulfolobus
Domain Bacteria Agrobacterium, Anabaena,
Bacillus, Escherichia,
Rhizobium, Salmonella,
Streptomyces

**Syllabus for the Experimental Component**

Please note:

□□The syllabus for National Standard Examination in Biology (NSEB) is broadly equivalent to the senior secondary level (Class XI and Class XII) of CBSE Biology. This is only a rough guideline, and there is no detailed syllabus given for NSEB.

□□The syllabus for Indian National Biology Olympiad (INBO) is broadly similar to NSEB but the difficulty level of the questions will be higher. Questions and problems in National Olympiads are usually nonconventional and of high difficulty level, comparable to International Olympiads.

# The International Olympiad in Informatics Syllabus

## 1 Version and status information

This is the Syllabus version intended for IOI 2020.

There have been no changes since the 2018 version.

The Syllabus is an official document related to the IOI. For each IOI, an up-to-date version of the Syllabus is produced by the ISC, as described in the IOI Regulations, Statue 3.13.

## 2 Authors and Contact Information

The original proposal of the IOI Syllabus was co-authored by Tom Verhoeff[1], Gyula Horváth[2], Krzysztof Diks[3], and Gordon Cormack[4].

Since 2007, the following people have maintained the syllabus and made significant contributions: Michal Forišek[5], Jakub Łącki[6], and Richard Peng[7].

The most recent batch of revisions to the Syllabus was made by the ISC between February and July 2016.

You are welcome to send any feedback on the Syllabus to the current maintainer's e-mail address (forisek@dcs.fmph.uniba.sk).

For people interested in contributing to the quality of the Syllabus, some additional background on the Syllabus and other miscellaneous information can be found at http://ksp.sk/~misof/ioi-syllabus/.

[1] TU Eindhoven, The Netherlands, t.verhoeff@tue.nl
[2] University of Szeged, Hungary, horvath@inf.u-szeged.hu
[3] Warsaw University, Poland, diks@mimuw.edu.pl
[4] University of Waterloo, Canada, gvcormac@uwaterloo.ca
[5] Comenius University, Slovakia, forisek@dcs.fmph.uniba.sk
[6] Warsaw University, Poland, j.lacki@mimuw.edu.pl
[7] Georgia Tech, USA, richard.peng@gmail.com

# 3 Introduction

During the years, the Syllabus has evolved. Currently it has the following purposes:

- It specifies a small set of required prerequisite knowledge. Below, this is given in the category "Included, unlimited" and to some extent also in "Included, to be defined".
- It serves as a set of guidelines that help decide whether a task is suitable for the International Olympiad in Informatics (IOI). Based on this document, the International Scientific Committee (ISC) evaluates the task proposals when selecting the competition tasks.
- As a consequence of the previous item, another purpose of the Syllabus is to help the organizers of national olympiads prepare their students for the IOI.

The Syllabus aims to achieve these goals by providing a classification of topics and concepts from mathematics and computer science. More precisely, this Syllabus classifies each topic into one of six categories. Ordered by topic suitability, these are:

| | |
|---|---|
| ✓ | Included, unlimited |
| ✓🗎 | Included, to be defined |
| ✓📄 | Included, not for task description |
| ? | Outside of focus |
| ✗❓ | Excluded, but open to discussion |
| ✗ | Explicitly excluded |

In the next section we explain the scope of each category.

## 4 Categories

This Syllabus classifies a selection of topics into six different categories. Obviously, such a set of topics can never be exhaustive. Instead, the list given in this Syllabus should serve as examples that map out the boundary. Topics not explicitly mentioned in the Syllabus should be classified as follows:

- Anything that is a prerequisite of an Included topic is also Included.
- Anything that is an extension of an Excluded topic or similar to an Excluded topic is also Excluded.
- Anything else that is not mentioned in the Syllabus is considered Outside of focus.

Note that issues related to the usage of suitable terminology and notations in competition tasks are beyond the scope of this document.[8]

If there is a particular topic for which you are not sure how it should be classified, we invite you to submit a clarification request to the current Syllabus maintainer.

Here are the definitions of the six possible classifications:

**✓ Included, unlimited**

Topics in this category are considered to be prerequisite knowledge. Contestants are expected to know them. These topics can appear in task descriptions without further clarification.

Example: *Integer* in §5.1

**✓🗎 Included, to be defined**

Contestants should know this topic, but when it appears in a task description, the statement should contain a sufficient definition. This category is usually applied in situations where a general concept that would be ✓ has many different "flavors" and a formal definition is needed to distinguish among those.

Example: *Directed graph* in §5.2 DS2

**✓📄 Included, not for task description**

Topics that belong to this category should not appear in tasks descriptions. However, developing solutions and understanding model solutions may require the knowledge of these topics.

[8] See T. Verhoeff: *Concepts, Terminology, and Notations for IOI Competition Tasks*, http://scienceolympiads.org/ioi/sc/documents/terminology.pdf

Example: *Asymptotic analysis of upper complexity bounds* in §6.2 AL1

Note: This is the main category that should be of interest when preparing contestants for the IOI. It should be noted that this set of topics contains a wide range of difficulties, starting from simple concepts and ending with topics that can appear in problems that aim to distinguish among the gold medallists. It is **not** expected that all contestants should know everything listed in this category.

**? Outside of focus**

Any topic that is not explicitly addressed by the Syllabus should be considered to belong to this category.

Contestants are not expected to have knowledge of these topics. Most competition tasks will not be related to any topics from this category.

However, this does not prevent the inclusion of a competition task that is related to a particular topic from this category. The ISC may wish to include such a competition task in order to broaden the scope of the IOI.

If such a task is considered for the IOI, the ISC will make sure that the task can reasonably be solved without prior knowledge of the particular topic, and that the task can be stated in terms of ✓ and ✓🗎 concepts in a precise, concise, and clear way.

Examples of such tasks being used at recent IOIs include:

- Languages (a.k.a. Wikipedia) from IOI 2010 in Canada
- Odometer (a.k.a. robot with pebbles) from IOI 2012 in Italy
- Art class from IOI 2013 in Australia.

**✗ Explicitly excluded**

Some of the harder algorithmic topics are explicitly marked as excluded. It is guaranteed that there will not be a competition task that *requires* the contestants to know these areas.

Furthermore, the tasks will be set with the goal that knowledge of Excluded topics should not help in obtaining simpler solutions / solutions worth more points.

This category contains topics whose inclusion will result in problems that are unaccessible to a significant portion of IOI participants. This includes but is not limited to hard textbook algorithms and advanced areas in mathematics.

Still, note that the Syllabus must not be interpreted to restrict in any way the techniques that contestants are allowed to apply in solving the competition tasks.

Examples: *Calculus* in §5.3

**✗❓ Excluded, but open to discussion**

As the Syllabus is a living document, there can be cases when we consider bringing in some of the Excluded topics. Usually, the topics in question are natural extensions of Included topics, or ones where drawing an exact boundary is difficult. Should such topics appear, they will be temporarily classified as "Excluded, but open to discussion", and by doing so we encourage all members of the IOI community to give us feedback on these topics.

The rest of this document contains the classification of topics.

# 5 Mathematics

## 5.1 Arithmetics and Geometry

✓ Integers, operations (incl. exponentiation), comparison

✓ Basic properties of integers (sign, parity, divisibility)

✓ Basic modular arithmetic: addition, subtraction, multiplication

✓🖹 Prime numbers

✓ Fractions, percentages

✓ Line, line segment, angle, triangle, rectangle, square, circle

✓ Point, vector, coordinates in the plane

✓ Polygon (vertex, side/edge, simple, convex, inside, area)

✓🗎 Euclidean distances

✓🖹 Pythagorean theorem

✗❓ Additional topics from number theory.

✗ geometry in 3D or higher dimensional spaces

✗ analyzing and increasing precision of floating-point computations

✗ modular division and inverse elements

- ✗ complex numbers,
- ✗ general conics (parabolas, hyperbolas, ellipses)
- ✗ trigonometric functions

## 5.2 Discrete Structures (DS)

### DS1. Functions, relations, and sets

- ✓🗎 Functions (surjections, injections, inverses, composition)
- ✓🗎 Relations (reflexivity, symmetry, transitivity, equivalence relations, total/linear order relations, lexicographic order)
- ✓🗎 Sets (inclusion/exclusion, complements, Cartesian products, power sets)

- ✗ Cardinality and countability (of infinite sets)

### DS2. Basic logic

- ✓ First-order logic
- ✓ Logical connectives (incl. their basic properties)
- ✓ Truth tables
- ✓ Universal and existential quantification (Note: statements should avoid definitions with nested quantifiers whenever possible.)
- ✓📄 Modus ponens and modus tollens

- ? Normal forms

- ✗ Validity
- ✗ Limitations of predicate logic

### DS3. Proof techniques

- ✓🗎 Notions of implication, converse, inverse, contrapositive, negation, and contradiction
- ✓📄 Direct proofs, proofs by: counterexample, contraposition, contradiction
- ✓📄 Mathematical induction
- ✓📄 Strong induction (also known as complete induction)
- ✓ Recursive mathematical definitions (incl. mutually recursive definitions)

### DS4. Basics of counting

- ✓ Counting arguments (sum and product rule, arithmetic and geometric progressions, Fibonacci numbers)
- ✓🖹 Permutations and combinations (basic definitions)
- ✓🖹 Factorial function, binomial coefficients
- ✓📄 Inclusion-exclusion principle
- ✓📄 Pigeonhole principle
- ✓📄 Pascal's identity, Binomial theorem

- ✗ Solving of recurrence relations
- ✗ Burnside lemma

### DS5. Graphs and trees

- ✓🖹 Trees and their basic properties, rooted trees
- ✓🖹 Undirected graphs (degree, path, cycle, connectedness, Euler/Hamilton path/cycle, handshaking lemma)
- ✓🖹 Directed graphs (in-degree, out-degree, directed path/cycle, Euler/Hamilton path/cycle)
- ✓🖹 Spanning trees
- ✓🖹 Traversal strategies
- ✓🖹 'Decorated' graphs with edge/node labels, weights, colors
- ✓🖹 Multigraphs, graphs with self-loops
- ✓🖹 Bipartite graphs
- ✓📄 Planar graphs

- ✗ Hypergraphs
- ✗ Specific graph classes such as perfect graphs
- ✗ Structural parameters such as treewidth and expansion
- ✗ Planarity testing
- ✗ Finding separators for planar graphs

### DS6. Discrete probability

Applications where everything is finite (and thus arguments about probability can be easily turned into combinatorial arguments) are **?**, everything more complicated is ✗.

### 5.3 Other Areas in Mathematics

- ✗ Geometry in three or more dimensions.
- ✗ Linear algebra, including (but not limited to):
  - Matrix multiplication, exponentiation, inversion, and Gaussian elimination
  - Fast Fourier transform
- ✗ Calculus
- ✗ Theory of combinatorial games, e.g., NIM game, Sprague-Grundy theory
- ✗ Statistics

# 6 Computing Science

## 6.1 Programming Fundamentals (PF)

### PF1. Fundamental programming constructs(for abstract machines)

- ✓ Basic syntax and semantics of a higher-level language (at least one of the specific languages available at an IOI, as announced in the *Competition Rules* for that IOI)
- ✓ Variables, types, expressions, and assignment
- ✓ Simple I/O
- ✓ Conditional and iterative control structures
- ✓ Functions and parameter passing
- ✓🖹 Structured decomposition

### PF2. Algorithms and problem-solving

- ✓🖹 Problem-solving strategies (understand–plan–do–check, separation of concerns, generalization, specialization, case distinction, working backwards, etc.)[9]
- ✓🖹 The role of algorithms in the problem-solving process
- ✓🖹 Implementation strategies for algorithms (also see §7 SE1)
- ✓🖹 Debugging strategies (also see §7 SE3)
- ✓🗎 The concept and properties of algorithms (correctness, efficiency)

[9] See G. Polya: *How to Solve It: A New Aspect of Mathematical Method*, Princeton Univ. Press, 1948

### PF3. Fundamental data structures

- ✓ Primitive types (boolean, signed/unsigned integer, character)
- ✓ Arrays (incl. multicolumn dimensional arrays)
- ✓ Strings and string processing
- ✓🗎 Static and stack allocation (elementary automatic memory management)
- ✓🗎 Linked structures
- ✓🗎 Implementation strategies for graphs and trees
- ✓🗎 Strategies for choosing the right data structure
- ✓🖹 Elementary use of real numbers in numerically stable tasks
- ✓🖹 The floating-point representation of real numbers, the existence of precision issues.[10]
- ✓🖹 Pointers and references

- ? Data representation in memory,
- ? Heap allocation,
- ? Runtime storage management,
- ? Using fractions to perform exact calculations.

- ✗ Non-trivial calculations on floating point numbers, manipulating precision errors

Regarding floating point numbers, there are well-known reasons why they should be, in general, avoided at the IOI.[11] However, the currently used interface removes some of those issues. In particular, it should now be safe to use floating point numbers in some types of tasks – e.g., to compute some Euclidean distances and return the smallest one.

### PF4. Recursion

- ✓ The concept of recursion
- ✓ Recursive mathematical functions

[10] Whenever possible, avoiding floating point computations completely is the preferred solution.

[11] See G. Horváth and T. Verhoeff: *Numerical Difficulties in Pre-University Education and Competitions*, Informatics in Education 2:21–38, 2003

- ✓ Simple recursive procedures (incl. mutual recursion)
- ✓🗎 Divide-and-conquer strategies
- ✓🗎 Implementation of recursion
- ✓🗎 Recursive backtracking

### PF5. Event-driven programming

Some competition tasks may involve a dialog with a reactive environment. Implementing such an interaction with the provided environment is ✓🖹.

Everything not directly related to the implementation of reactive tasks is **?**.

## 6.2 Algorithms and Complexity (AL)

We quote from the IEEE-CS Curriculum:

> Algorithms are fundamental to computer science and software engineering. The real-world performance of any software system depends only on two things: (1) the algorithms chosen and (2) the suitability and efficiency of the various layers of implementation. Good algorithm design is therefore crucial for the performance of all software systems. Moreover, the study of algorithms provides insight into the intrinsic nature of the problem as well as possible solution techniques independent of programming language, programming paradigm, computer hardware, or any other implementation aspect.

### AL1. Basic algorithmic analysis

- ✓🖹 Algorithm specification, precondition, postcondition, correctness, invariants
- ✓🗎 Asymptotic analysis of upper complexity bounds (informally if possible)
- ✓🗎 Big O notation
- ✓🗎 Standard complexity classes: constant, logarithmic, linear, $\mathcal{O}(n \log n)$, quadratic, cubic, exponential, etc.
- ✓🗎 Time and space tradeoffs in algorithms
- ✓🗎 Empirical performance measurements.

- **?** Identifying differences among best, average, and worst case behaviors,

- ? Little o, Omega, and Theta notation,
- ? Tuning parameters to reduce running time, memory consumption or other measures of performance

- ✗ Asymptotic analysis of average complexity bounds
- ✗ Using recurrence relations to analyze recursive algorithms

### AL2. Algorithmic strategies

- ✓📄 Simple loop design strategies
- ✓📄 Brute-force algorithms (exhaustive search)
- ✓📄 Greedy algorithms
- ✓📄 Divide-and-conquer
- ✓📄 Backtracking (recursive and non-recursive), Branch-and-bound
- ✓📄 Dynamic programming[12]

- ? Heuristics
- ? Finding good features for machine learning tasks[13]
- ? Discrete approximation algorithms
- ? Randomized algorithms.

- ✗ Clustering algorithms (e.g. $k$-means, $k$-nearest neighbor)
- ✗ Minimizing multi-variate functions using numerical approaches.

### AL3a. Algorithms

- ✓📄 Simple algorithms involving integers: radix conversion, Euclid's algorithm, primality test by $\mathcal{O}(\sqrt{n})$ trial division, Sieve of Eratosthenes, factorization (by trial division or a sieve), efficient exponentiation
- ✓📄 Simple operations on arbitrary precision integers (addition, subtraction, simple multiplication)[14]
- ✓📄 Simple array manipulation (filling, shifting, rotating, reversal, resizing, minimum/maximum, prefix sums, histogram, bucket sort)

[12] The IEEE-CS Curriculum puts this under AL8, but we believe it belongs here.

[13] E.g., finding a good way to classify images in the IOI 2013 Art class problem.

[14] The necessity to implement these operations should be obvious from the problem statement.

- ✓🖹 Simple string algorithms (e.g., naive substring search)
- ✓🖹 sequential processing/search and binary search
- ✓🖹 Quicksort and Quickselect to find the $k$-th smallest element.
- ✓🖹 $\mathcal{O}(n \log n)$ worst-case sorting algorithms (heap sort, merge sort)
- ✓🖹 Traversals of ordered trees (pre-, in-, and post-order)
- ✓🖹 Depth- and breadth-first traversals
- ✓🖹 Applications of the depth-first traversal tree, such as topological ordering and Euler paths/cycles
- ✓🖹 Finding connected components and transitive closures.
- ✓🖹 Shortest-path algorithms (Dijkstra, Bellman-Ford, Floyd-Warshall)
- ✓🖹 Minimum spanning tree (Jarník-Prim and Kruskal algorithms)
- ✓🖹 $O(VE)$ time algorithm for computing maximum bipartite matching.
- ✓🖹 Biconnectivity in undirected graphs (bridges, articulation points).
- ✓🖹 Connectivity in directed graphs (strongly connected components).
- ✓🖹 Basics of combinatorial game theory, winning and losing positions, minimax algorithm for optimal game playing

- ✗❓ Maximum flow. Flow/cut duality theorem.

- ✗ Optimization problems that are easiest to analyze using matroid theory. Problems based on matroid intersecions (except for bipartite matching).
- ✗ Lexicographical BFS, maximum adjacency search and their properties

### AL3b. Data structures

- ✓🗎 Stacks and queues
- ✓🖹 Representations of graphs (adjacency lists, adjacency matrix)
- ✓🗎 Binary heap data structures
- ✓🖹 Representation of disjoint sets: the Union-Find data structure.

- ✓🗎 Statically balanced binary search trees. Instances of this include binary index trees (also known as Fenwick trees) and segment trees (also known as interval trees and tournament trees).[15]
- ✓🗏 Balanced binary search trees[16]
- ✓🗎 Augmented binary search trees
- ✓🗏 $O(\log n)$ time algorithms for answering lowest common ancestor queries in a static rooted tree.[17]
- ✓🗏 Creating persistent data structures by path copying.
- ✓🗏 Nesting of data structures, such as having a sequence of sets.
- ✓🗎 Tries

- ✗❓ String algorithms and data structures: there is evidence that the inter-reducibility between KMP, Rabin-Karp hashing, suffix arrays/tree, suffix automaton, and Aho-Corasick makes them difficult to separate.
- ✗❓ Heavy-light decomposition and separator structures for static trees.
- ✗❓ Data structures for dynamically changing trees and their use in graph algorithms.

- ✗ Complex heap variants such as binomial and Fibonacci heaps,
- ✗ Using and implementing hash tables (incl. strategies to resolve collisions)
- ✗ Two-dimensional tree-like data structures (such as a 2D statically balanced binary tree or a treap of treaps) used for 2D queries.
- ✗ Fat nodes and other more complicated ways of implementing persistent data structures.

### AL4. Distributed algorithms

This entire section is **?**.

[15] Not to be confused with similarly-named data structures used in computational geometry.

[16] Problems will not be designed to distinguish between the implementation of BBSTs, such as treaps, splay trees, AVL trees, or scapegoat trees

[17] Once again, different implementations meeting this requirement will not be distinguished.

### AL5. Basic computability

All topics related to computability are ✗. This includes the following: Tractable and intractable problems; Uncomputable functions; The halting problem; Implications of uncomputability.

However, see AL7 for basic computational models.

### AL6. The complexity classes P and NP

Topics related to non-determinism, proofs of NP-hardness (reductions), and everything related is ✗.

Note that this section only covers the results usually contained in undergraduate and graduate courses on formal languages and computational complexity. The classification of these topics as ✗ does not mean that an NP-hard problem cannot appear at an IOI.

### AL7. Automata and grammars

✓🖹 Understanding a simple grammar in Backus-Naur form

**?** Formal definition and properties of finite-state machines,
**?** Context-free grammars and related rewriting systems,
**?** Regular expressions

✗ Properties other than the fact that automata are graphs and that grammars have parse trees.

### AL8. Advanced algorithmic analysis

✓📄 Amortized analysis.

**?** Online algorithms
**?** Randomized algorithms

✗ Alpha-beta pruning

### AL9. Cryptographic algorithms

This entire section is **?**.

## AL10. Geometric algorithms

In general, the ISC has a strong preference towards problems that can be solved using integer arithmetics to avoid precision issues. This may include representing some computed values as exact fractions, but extensive use of such fractions in calculations is discouraged.

Additionally, if a problem uses two-dimensional objects, the ISC prefers problems in which such objects are rectilinear.

- ✓🗎 Representing points, vectors, lines, line segments.
- ✓🗎 Checking for collinear points, parallel/orthogonal vectors and clockwise turns (for example, by using dot products and cross products).
- ✓🗎 Intersection of two lines.
- ✓🗎 Computing the area of a polygon from the coordinates of its vertices.[18]
- ✓🗎 Checking whether a (general/convex) polygon contains a point.
- ✓🗎 Coordinate compression.
- ✓🗎 $\mathcal{O}(n \log n)$ time algorithms for convex hull
- ✓🗎 Sweeping line method

- ✗ Point-line duality
- ✗ Halfspace intersection, Voronoi diagrams, Delaunay triangulations.
- ✗ Computing coordinates of circle intersections against lines and circles.
- ✗ Linear programming in 3 or more dimensions and its geometric interpretations.
- ✗ Center of mass of a 2D object.
- ✗ Computing and representing the composition of geometric transformations if the knowledge of linear algebra gives an advantage.

## AL11. Parallel algorithms

This entire section is **?**.

[18] The recommended way of doing so is to use cross products or an equivalent formula. TODO url

### 6.3 Other Areas in Computing Science

Except for GV (specified below), all areas are ✗.

**AR. Architecture and Organization**

**OS. Operating Systems**

**NC. Net-Centric Computing** (a.k.a. cloud computing)

**PL. Programming Languages**

**HC. Human-Computer Interaction**

**GV. Graphics and Visual Computing**

Basic aspects of processing graphical data are **?**, everything else (including the use of graphics libraries such as OpenGL) is ✗.

**IS. Intelligent Systems**

**IM. Information Management**

**SP. Social and Professional Issues**

**CN. Computational Science**

Notes: AR is about digital systems, assembly language, instruction pipelining, cache memories, etc. OS is about the *design* of operating systems, not their usage. PL is about the *analysis and design* of programming languages, not their usage. HC is about the *design* of user interfaces.

*Usage* of the operating system, GUIs and programming languages is covered in §8 and §6.1.

## 7 Software Engineering (SE)

We quote from the IEEE-CS Curriculum:

> Software engineering is the discipline concerned with the application of theory, knowledge, and practice for effectively and efficiently building software systems that satisfy the requirements of users and customers.

In the IOI competition, the application of software engineering concerns the use of light-weight techniques for small, one-off, single-developer projects under time pressure. All included topics are ✓📄.

## SE1. Software design

- ✓📄 Fundamental design concepts and principles
- ✓📄 Design patterns
- ✓📄 Structured design

In particular, contestants may be expected to

- – Transform an abstract algorithm into a concrete, efficient program expressed in one of the allowed programming languages, possibly using standard or competition-specific libraries.
- – Make their programs read data from and write data to text files according to a prescribed simple format

- ✗ Software architecture,
- ✗ Design for reuse,
- ✗ Object-Oriented analysis and design,
- ✗ Component-level design

## SE2. Using APIs

- ✓📄 API (Application Programming Interface) programming

In particular, contestants may be expected to

- – Use competition-specific libraries according to the provided specification.

- ✗ Programming by example,
- ✗ Debugging in the API environment,
- ✗ Class browsers and related tools,
- ✗ Introduction to component-based computing

## SE3. Software tools and environments

- ✓📄 Programming environments, incl. IDE (Integrated Development Environment)

In particular, contestants may be expected to

- Write and edit program texts using one of the provided program editors.
- Compile and execute their own programs.
- Debug their own programs.

✗ Testing tools,<br>
✗ Configuration management tools<br>
✗ Requirements analysis and design modeling tools,<br>
✗ Tool integration mechanisms

### SE4. Software processes

✓📄 Software life-cycle and process models

In particular, contestants may be expected to

- Understand the various phases in the solution development process and select appropriate approaches.

✗ Process assessment models,<br>
✗ Software process metrics

### SE5. Software requirements and specification

✓📄 Functional and nonfunctional requirements

✓📄 Basic concepts of formal specification techniques

In particular, contestants may be expected to

- Transform a precise natural-language description (with or without mathematical formalism) into a problem in terms of a computational model, including an understanding of the efficiency requirements.

✗ Prototyping,<br>
✗ Requirements elicitation,<br>
✗ Requirements analysis modeling techniques

### SE6. Software validation

- ✓🗎 Testing fundamentals, including test plan creation and test case generation
- ✓🗎 Black-box and white-box testing techniques
- ✓🗎 Unit, integration, validation, and system testing
- ✓🗎 Inspections

In particular, contestants may be expected to

- – Apply techniques that maximize the the opportunity to detect common errors (e.g. through well-structured code, code review, built-in tests, test execution).
- – Test (parts of) their own programs.

- ✗ Validation planning,
- ✗ Object-oriented testing

### SE7. Software evolution

- ✗ Software maintenance,
- ✗ Characteristics of maintainable software,
- ✗ Re-engineering,
- ✗ Legacy systems,
- ✗ Software reuse

### SE8. Software project management

- ✓🗎 Project scheduling (especially time management)
- ✓🗎 Risk analysis
- ✓🗎 Software configuration management

In particular, contestants may be expected to

- – Manage time spent on various activities.
- – Weigh risks when choosing between alternative approaches.
- – Keep track of various versions and their status while developing solutions.

- ✗ Software quality assurance,
- ✗ Team management,
- ✗ Software measurement and estimation techniques,
- ✗ Project management tools

### SE9. Component-based computing

This entire section is ✗.

### SE10. Formal methods

✓📄 Formal methods concepts (notion of correctness proof, invariant)

✓📄 Pre and post assertions

In particular, contestants may be expected to

- Reason about the correctness and efficiency of algorithms and programs.

✗ Formal verification,

✗ Formal specification languages,

✗ Executable and non-executable specifications

### SE11. Software reliability

This entire section is ✗.

### SE12. Specialized systems development

This entire section is ✗.

## 8 Computer Literacy

The text of this section is ✓📄.

Contestants should know and understand the basic structure and operation of a computer (CPU, memory, I/O). They are expected to be able to use a standard computer with graphical user interface, its operating system with supporting applications, and the provided program development tools for the purpose of solving the competition tasks. In particular, some skill in file management is helpful (creating folders, copying and moving files).

Details of these facilities will be stated in the *Competition Rules* of the particular IOI. Typically, some services are available through a standard web browser. Possibly, some competition-specific tools are made available, with separate documentation.

It is often the case that a number of equivalent tools are made available. The contestants are not expected to know all the features of all these tools. They can make their own choice based on what they find most appropriate.

The following topics are all **?**: Calculator, Word-processors, Spreadsheet applications, Database management systems, E-mail clients, Graphics tools (drawing, painting).